Model Repair via Symmetry

Paul Attie* and William Cocke†

April 26, 2022

Abstract

The symmetry of a Kripke structure $M$ has been exploited to replace a model check of $M$ by a model check of the potentially smaller structure $N$ obtained as the quotient of $M$ by its symmetry group $G$. We extend previous work to model repair: identify a substructure that satisfies a given temporal logic formula. We show that the substructures of $M$ that are preserved by $G$ form a lattice that maps to the substructure lattice of $N$. We also show the existence of a monotone Galois connection between the lattice of substructures of $N$ and the lattice of substructures of $M$ that are "maximal" w.r.t. an appropriately defined group action of $G$ on $M$. These results enable us to repair $N$ and then to lift the repair to $M$. We can thus repair symmetric finite-state concurrent programs by repairing the corresponding $N$, thereby effecting program repair while avoiding state-explosion.

1 Introduction

The model checking [9] problem for (finite-state) concurrent programs $P = P_1 \cdots P_n$ w.r.t. a temporal logic formula $f$ is to verify that the Kripke structure $M$ that is generated by the execution of $P$ satisfies $f$. A major obstacle has been state explosion: in general, the size of $M$ is exponential in the number of processes $n$. When $M$ and $f$ both have a high degree of symmetry in the process index set $\{1, \ldots, n\}$, the use of symmetry reduction to ameliorate state-explosion can yield significant reduction in the complexity of model checking $M \models f$.

An approach that has been thoroughly investigated is the computation of the quotient structure $\overline{M} = M/G$ of $M$ by the symmetry group $G$ of symmetries of both $M$ and $f$. $\overline{M}$ can be computed directly from $P$, thereby avoiding the expensive computation of the large structure $M$. Since model checking $\overline{M} \models f$ is linear in the size of $\overline{M}$ [9], this provides significant savings if $\overline{M}$ is small. A related approach is the use of structural methods to express symmetric designs, e.g., parameterized systems, where processes are all instances of a common template (possibly with a distinguished controller process), and rings of processes, where all communication is between a process and its neighbors in the ring.

*School of Computer and Cyber Sciences, Augusta University, Augusta GA
†Corresponding author, School of Computer and Cyber Sciences, Augusta University, Augusta GA, wcocke@augusta.edu
1.1 Our contributions

We present a theory of substructures of Kripke structures which provides a correspondence theorem between certain substructures of the Kripke structure $M$ and the substructures of the quotient structure $\overline{M}$. Substructures of $M$ correspond to possible repairs of $M$, while substructures of $\overline{M}$ correspond to repairs of $M$. Because the correspondence between the two lattices of substructures preserves semantic valuation of certain formulae, we can associate certain repairs of $M$ with repairs of $\overline{M}$. Syntactically the correspondence of Kripke substructures lattices is of independent mathematical interest as an example of a monotone Galois connection.

We then build on this theory to extend symmetry-based model checking to concurrent program repair: given a concurrent program $P$ that may not satisfy $f$, modify $P$ to produce a program that does satisfy $f$. Given $P, f$, and a group $G$ of symmetries of both $P$ and $f$, our method directly computes the quotient $M/G$ (following [26]), then repairs $M/G$, using the algorithm of [2], and finally, extracts a correct program from the repaired structure.

2 Related Work

As discussed above, group theoretic approaches to symmetry-reduction [22, 20, 27, 8, 28, 29, 10, 30, 11, 17, 19, 33, 16, 25, 21] compute the quotient $M/G$ and model check that, instead of the original (much larger) structure $M$. A key consideration is to devise a symmetry group $G$ that is as large as possible, so as to achieve the greatest possible reductions. Parametrized systems [23, 24, 17, 18, 25, 21, 16, 13, 19] are those which consist of an arbitrary number of isomorphic instances of a single process template. The usual approach is to establish a cutoff: a small value $n$ such that correctness of a system with $n$ instances implies correctness of arbitrarily large systems. Special subclasses of parametrized systems include token-passing systems [1, 12], which circulate one or more tokens amongst isomorphic processes, and systems based on rendezvous/broadcast/multicast [32, 31, 20]. Bloem et al. [3] surveys the decidability of verification problems for parametrized systems.

Buccafurri et al. [5] posed the repair problem for CTL and solved it using abductive reasoning to generate repair suggestions that are verified by model checking. Jobstmann et al. [34] and Staber et al. [35] present game-based repair methods for programs and circuits. Their method is complete for invariants only. Carrillo and Rosenblueth [6] presents a syntax-directed repair method that uses “protections” to deal with conjunction, and a representation of Kripke structures as “tables”. Chatzieleftheriou et al. [7] repair abstract structures, using Kripke modal transition systems and 3-valued CTL semantics. von Essen and Jobstmann [36] present a game-based repair method which attempts to keep the repaired program close to the original faulty program, by also specifying a set of traces that the repair must leave intact.

3 Preliminary Definitions

In this section we include the formal definitions necessary for structure repair via symmetry. The section is broken into two parts: the first part covers the
basic definitions of Kripke structures and CTL as can be found in [15]; the second part covers the basic connections with group theory.

3.1 Kripke Structures and CTL

Given a set \(AP\) of atomic propositions we define the propositional branching-time temporal logic CTL\([13, 15]\) by the following grammar:

\[
\varphi ::= \text{true} | \text{false} | p | \neg \varphi | \varphi \land \varphi | \varphi \lor \varphi | AX\varphi | EX\varphi | A[\varphi R \varphi] | E[\varphi R \varphi]
\]

where \(p \in AP\), and \(\text{true}, \text{false}\) are constant propositions whose interpretation is the semantic truth values \(tt, ff\) respectively. A propositional formula is a CTL formula not containing the \(AX, EX, AR, ER\) modalities. The semantics of CTL are defined with respect to a Kripke structure:

**Definition 3.1** (Kripke structure). A Kripke structure \(M\) is a tuple \((S, S_0, T, L, AP)\) where \(S\) is a finite nonempty set of states, \(S_0 \subseteq S\) is a nonempty set of initial states, \(T \subseteq (S \times S)\) is a transition relation, \(AP\) is a set of atomic propositions, and \(L : S \rightarrow 2^{AP}\) is a labeling function that associates each state \(s \in S\) with a subset of atomic propositions, namely those that hold in state \(s\). We require that \(M\) be total, i.e., \(\forall s \in S, \exists t \in S : (s, t) \in T\).

When referring to the constituents of \(M = (S, S_0, T, L, AP)\), we write \(M_S, M_{S_0}, M_T, M_L,\) and \(M_{AP}\) respectively. State \(t\) is a successor of state \(s\) in \(M\) iff \((s, t) \in T\). A path \(\pi\) in \(M\) is a (finite or infinite) sequence of states, \(\pi = s_0, s_1, \ldots\), such that \(\forall i \geq 0 : (s_i, s_{i+1}) \in T\). A fullpath is an infinite path. State \(t\) is reachable from state \(s\) iff there is a path from \(s\) to \(t\). State \(t\) is reachable iff there is a path from an initial state to \(t\). Without loss of generality, we assume that a Kripke structure does not contain unreachable states, i.e., every \(s \in S\) is reachable.

Throughout the rest of the paper we will refer to two running examples of Kripke structures. We will refer to these as Example 2-mutex and Example Box.

**Example 3.2** (2-mutex). This is the classic 2-mutex example. Our Kripke structure corresponds to two processes acting concurrently. Each process can be in one of 3 states, either Neutral, Trying, or Critical and can transit from Neutral to Trying, from Trying to Critical, and from Critical to Neutral.

We represent this Kripke structure diagrammatically in the left hand side of Figure 5. We see all of the transitions represented by arrows, the initial states are shaded green, and the labels can be seen in each state.

**Example 3.3** (Box). The so-called Box Kripke structure in Figure 7 is an example of a highly symmetric structure, as we will discuss in section 5. This structure has 4 states and is represented in Figure 1. Of note, we assign all states with the same label, making Box a boring structure with respect to semantic evaluation. However, the simplicity of Box will be useful when considering the syntactic concept of a group of state mappings and the various substructures of Box.
A CTL formula $\varphi$ is evaluated (i.e., is true or false) in a state $s$ of a Kripke structure $\mathcal{M}$ [15]. Its value in $s$ depends on the value of its subformulae in states that are reachable from $s$. Write $\mathcal{M}, s \models \varphi$ when $s$ is true in state $s$ of structure $\mathcal{M}$, and write $\mathcal{M} \models \varphi$ to abbreviate $\forall s_0 \in S_0 : \mathcal{M}, s_0 \models \varphi$, i.e., $\varphi$ holds in all initial states of $\mathcal{M}$. Appendix A gives the formal definition of $\models$, which proceeds by induction on the structure of CTL formulae.

$[\varphi \mathcal{R} \psi]$ ($\varphi$ “releases” $\psi$) means that $\psi$ must hold up to and including the first state where $\varphi$ holds, and forever if $\varphi$ never holds. We introduce the abbreviations $A[\varphi U \psi]$ for $\neg E[\neg \varphi \mathcal{R} \neg \psi]$, $E[\varphi U \psi]$ for $\neg A[\neg \varphi \mathcal{R} \neg \psi]$, $A F \varphi$ for $A[\text{true} U \varphi]$, $E F \varphi$ for $E[\text{true} U \varphi]$, $A G \varphi$ for $A[\text{false} R \varphi]$, $E G \varphi$ for $E[\text{false} R \varphi]$. We use $tt$, $ff$ for the (semantic) truth values of true, false, respectively. Whereas true, false are atomic propositions whose interpretation is always $tt$, $ff$, respectively.

### 4 Substructures of Kripke structures

Symmetry-based repairs are motivated by the correspondence between certain substructures of two related Kripke structures which we introduce as Theorem 5.16. A substructure of a Kripke structure is defined as follows:

**Definition 4.1** (Substructure). Given Kripke structures $\mathcal{M}$ and $\mathcal{M}'$, we say that $\mathcal{M}'$ is a substructure of $\mathcal{M}$, denoted $\mathcal{M}' \leq \mathcal{M}$, iff the following all hold:

1. $\mathcal{M}'_S \subseteq \mathcal{M}_S$.
2. $\mathcal{M}'_{S_0} = \mathcal{M}_{S_0} \cap \mathcal{M}'_S$.
3. $\mathcal{M}'_T \subseteq \mathcal{M}_T$.
4. $\mathcal{M}'_{AP} = \mathcal{M}_{AP}$.
5. $\mathcal{M}'_L = \mathcal{M}_L | S'$ (where $|$ denotes domain restriction).

We require substructures to be total. To reiterate, a substructure $\mathcal{N}$ of a Kripke structure $\mathcal{M}$ is total. For mathematical necessity in what follows, we allow for the ‘empty’ substructure. Since substructures must be total, some Kripke structures contain only two substructures, i.e., the empty substructure and the whole structure.

We return briefly to our examples.

**Example 4.2** (Example 2-mutex). Let $\mathcal{M}$ be the Kripke structure correspond to our running example 2-mutex (see Figure 5).

We can form a number of substructures of $\mathcal{M}$. For example, there is a substructure of $\mathcal{M}$ consisting only of states $s_0, s_1, s_2, s_3$ and $s_5$ along with all
edges from $M$. This substructure corresponds to computations wherein each process must return to neutral before the other process can enter the trying state. The states $s_0, s_1, s_3$ also form a substructure.

**Example 4.3 (Example Box).** Like our 2-mutex example, Example Box has many substructures. We present some of these substructures in Figure 2.

As seen in the above examples, even a simple Kripke structure can have many different substructures, even on the same set of states.

We can define a partially ordered set of substructures of a Kripke structure. This partially ordered set is a lattice i.e., it has a join and a meet operation. Before we define this lattice, we investigate a few more basic properties regarding substructures of Kripke structures.

Given a Kripke structure $M$, a substructure $M'$ of $M$, and a substructure $M''$ of $M'$ it is immediate that $M''$ is a substructure of $M'$.

**Lemma 4.4.** Let $M$ be a Kripke structure and suppose that $N$ and $N'$ are substructures of $M$. Then $\bigvee N \land N' = (N_S \cup N'_S, N_S \cup N'_S, N'_R, N'_R, M_L | (N_S \cup N'_S), M_X)$ is the smallest substructure of $M$ containing both $N$ and $N'$.

**Proof.** We will write $N_y$ for $N \land N'$. By construction $N_y$ contains both $N$ and $N'$. Let $T$ be any substructure of $M$ containing both $N$ and $N'$. Clearly $(N_y)_T \subseteq T_S, (N_y)_S \subseteq T_S, T_R \subseteq T_R$, and $T_L$ extends $(N_y)_L$.

It remains to show that $N_y$ is a substructure, i.e., that $N_y$ is total. Given a state $s_i$ in $(N_y)_T$, we know that either $s_i \in N_S$ or $s_i \in N'_S$. If $s_i \in N_S$, then there is an $t \in N_S$ such that $(s, t) \in N_T \subseteq (N_y)_T$. By symmetry if $s_i \in N'_S$ then there is an $t \in N'_S$ with $(s, t) \in (N_y)_T$. 

Given a finite set $X = \{X_0, X_1, \ldots, X_n\}$ of substructures of $M$, we can define the structure $\bigvee X = X_0 \lor X_1 \lor \cdots \lor X_n$.

**Lemma 4.5.** Let $M$ be a Kripke structure and suppose that $N$ and $N'$ are substructures of $M$. Then there is a largest substructure of $M$ contained in both $N$ and $N'$.

**Proof.** The empty structure is contained in both $N$ and $N'$. Let $X$ be the collection of structures contained in both $N$ and $N'$. Then $\bigvee X$ is contained in both $N$ and $N'$ and is the largest substructure of $M$ contained in both $N$ and $N'$.
Definition 4.6 (Join, Meet of substructures). Let \(N\) and \(N'\) be two substructures of \(M\). The join of \(N\) and \(N'\), written \(N \lor N'\), is the smallest substructure of \(M\) containing both \(N\) and \(N'\). The meet of \(N\) and \(N'\), written \(N \land N'\), is the largest substructure of \(M\) contained in both \(N\) and \(N'\).

We note that for two substructures \(N\) and \(N'\) of \(M\), the join \(N \lor N'\) has the simple description as given in Lemma 4.4. However, the meet \(N \land N'\), while well-defined, does not have such a simple description. It is possible that for two substructure \(N\) and \(N'\) of a Kripke structure \(M\), there are no substructures contain in both \(N\) and \(N'\). Hence the largest substructure contained in both \(N\) and \(N'\) could be empty.

We can now define a lattice of substructures \(\Lambda_M\) for a given structure \(M\).

Definition 4.7 (Lattice of Substructures). Given a Kripke structure \(M\) the lattice of substructures of \(M\) is

\[
\Lambda_M = (\{N : N \text{ is a substructure of } M\}, \leq),
\]

where the meet and join in \(\Lambda_M\) are the meet and join defined for substructures earlier.

There is another lattice of substructures related to \(M\). Given a subset \(S' \subseteq MS\), we can define the substructure \(N = \langle S' \rangle\) of \(M\) generated by \(S'\) as the largest substructure \(N'\) of \(M\) such that \(N' \subseteq S'\), equivalently

\[
N = \bigvee_{N' \subseteq M, \ N' \leq_M N'} N'.
\]

5 Quotient structures.

Broadly speaking, a symmetry of an object captures the concept of what we mean for two objects to be equal. For Kripke structures, we are in general not interested in full symmetry. Instead, we are interested in a concept of partial symmetry that allows for certain computations to take place. The reason for this is because true symmetry of a Kripke structure would have to act on not only the underlying state-space, but the set of atomic propositions as well. We introduce a related concept, which we call a state-mapping. Some authors call a ‘state-mapping’ a ‘symmetry’ of \(M\). Due to the mathematical meaning of the word symmetry, we do not follow this tradition.

5.1 Groups Acting on Kripke Structures

Definition 5.1. A state-mapping of \(M\) is a graph isomorphism of the state-space of \(M\) such that its restriction to the initial states is also an isomorphism, e.g., takes initial states to initial states. Formally, for a Kripke structure \(M\), a state-mapping of \(M\) is a bijection \(f : MS \to MS\) such that:

- \(f(MS_0) = MS_0\);
- For states \(s, t \in MS\) we have that \((s, t) \in MT \iff (f(s), f(t)) \in MT\).

The set of all state-mappings of \(M\) forms a group.
Definition 5.2 (G-closed). For a group G of state-mappings of a Kripke structure M, a substructure N of M is called G-closed, if G is a group of state-mappings of N, i.e., for every g ∈ G and s ∈ N, we have g(s) ∈ N.

Lemma 5.3. Let M be a Kripke structure and let G be a group of state mappings of M. Let N,N′ be two G-closed substructures of M. Then N ∨ N′ and N ∧ N′ are both G-closed.

Proof. We first show that N ∨ N′ is G-closed. Since (N ∨ N′)S = NS ∪ N′S, the group G acts on the states of N ∨ N′. We will show that every element of G is a state-mapping on N ∨ N′.

Let s, t ∈ (N ∨ N′)S. If g(s) = g(t) then s = t since g is bijection on M. Similarly, for any s ∈ (N ∨ N′)S, since g−1(s) ∈ (N ∨ N′)S.

If (s, t) ∈ (N ∨ N′)T, then (s, t) is in one of NT or N′T. In either case, (g(s), g(t)) is contained in the same set of transitions, and thus contained in (N ∨ N′)T.

We now consider N ∩ N′. Recall that N ∩ N′ is the largest substructure contained in both N and N′. If s ∈ (N ∩ N′)S, and g ∈ G, then there is a substructure of N and N′ containing g(s). Moreover as the largest substructure of N and N′, N ∩ N′ must contain this substructure, hence g(s) ∈ N ∩ N′.

Consider (s, t) ∈ (N ∩ N′)T. Since G is a group of state mappings of N ∩ N′ there is a substructure of N and N′ containing (g(s), g(t)). Since N ∩ N′ is the join of all substructures contained in N and N′, we conclude that it must contain (g(s), g(t)) as well.

By Lemma 5.3 we see that the G-closed substructures of M forms a sub-lattice of ΛM.

Definition 5.4 (Lattice of G-closed substructures). Given a Kripke structure M and a group G of state mappings of M, the poset of G-closed substructures of M forms a lattice. We call this lattice the lattice of G-closed substructures of M and write it as ΛM,G.

Example 5.5 (Example Box). Let M be Example Box, i.e., the Kripke structure described in Example 5.3. Let g be the map that simultaneously switches s1 and s2, and switches t1 and t2, i.e. g(s1) = s2, g(s2) = s1, g(t1) = t2, g(t2) = t1. Let G be the group consisting of g and the identity map on M. We note that G is not the entire group of state-mappings of M. The structure M has 10 G-closed substructures, including the empty structure. We present these structures in Figure 5.

As evidenced by Figure 3 in Example 5.5 a single Kripke structure can have many G-closed substructures.

5.2 Constructing the Quotient Model

Give a group G of state-mappings of a model M, we want to construct a quotient model ML. However, as noted, state-mappings do not contain any information about ML. To remedy this situation, we need a function that assigns a representative to each orbit of G.

Definition 5.6 (Representative map). Let M be a Kripke structure and suppose that G is a group of state-mappings of G. A representative map of M with respect to G is a function ϑG : M → M satisfying the following:
• For all $s, s' \in M_S$, if there is a $g \in G$ such that $g(s) = s'$ then $\vartheta_G(s) = \vartheta_G(s')$. (respects orbits)

• For all $s, s' \in M_S$, if there is no $g \in G$ such that $g(s) = s'$ then $\vartheta_G(s) \neq \vartheta_G(s')$. (separates orbits)

• For all $s \in M_S$, we have that $\vartheta_G(\vartheta_G(s)) = \vartheta_G(s)$, i.e., each orbit has a stable representative. (idempotent)

We are now ready to define the quotient structure.

**Definition 5.7 (Quotient structure).** Given a Kripke structure $M$, a group $G$ of state-mappings of $M$, and a representative map $\vartheta_G$ of $M$ with respect to $G$, we define the quotient structure $M/\vartheta_G$ of $M$ with respect to $G$ and $\vartheta_G$ as follows:

The set of states of $M/(G, \vartheta_G)$ consists of the image of $\vartheta_G$. We borrow the quotient notation from group theory and write $\overline{M}$ for $M/(G, \vartheta_G)$ and $\overline{\vartheta}(s')$ for $\vartheta_G(s)$.

We have $(\overline{s}, \overline{t}) \in \overline{M_T}$ if and only if there is some $s' \in M_S$ with $\vartheta_G(s') = s$ and some $t' \in M_S$ with $\vartheta_G(t') = t$ such that $(s', t') \in M_T$. 

Figure 3: The nonempty $G$-closed substructure of Example Box. Where $G$ is the group generated by the simultaneous swapping of indexes of both the $s_i$ and the $t_i$. 




An initial state of $\overline{M}$ is the image under $\vartheta_G$ of an initial state of $M$.

We define $\overline{M}_L : \overline{M}_S \rightarrow 2^{AP}$ by $\overline{M}_L(\overline{s}) = M_L(\vartheta_G(s))$.

$\overline{M}_{AP} = M_{AP}$.

We note that the states of a quotient structure correspond exactly to the orbits of states of the original structure. For transitions, we have a slightly more subtle correspondence. Consider the following example:

**Example 5.8** (Example Box). Let $M$ and $G$ be as in Example 5.5. Let $\vartheta_G$ be defined by $\vartheta(x_1) = x_1$ for $x \in \{s, t\}$. Then the quotient structure $\overline{M}/(G, \vartheta_G)$ has exactly $2$ states, $s_1 = \vartheta_G(s_1) = \vartheta_G(s_2)$ and $t_1 = \vartheta_G(t_1) = \vartheta_G(t_2)$ with transitions $(s_1, t_1), (t_1, s_1)$. Every $G$-closed substructure $\mathcal{N}$ of $M$ in Figure 3 will map to this structure via $\mathcal{N} \rightarrow \mathcal{N}/(G, \vartheta_G)$.

Many of these substructures do no have $(s_1, t_1)$ as a transition at all.

As shown in Example 5.8, the existence of a transition in the quotient structure, does not imply the existence of “the same” transition in the original structure. However, it does imply the existence of a corresponding transition in the original structure. Hence it is true that for every path through a quotient structure there is a corresponding path through the original Kripke structure. This property is what allows for model checking of the original structure via model checking of the quotient. This path correspondence is formalized in the below theorem which has appeared in many texts. See for example the work by Emerson and Sistla [26 3.1].

**Theorem 5.9** (Path Correspondence Theorem). Let $M$ be a Kripke structure and let $G$ be a group of state-mappings of $M$. Let $\vartheta_G$ be any representative map of $M$ with respect of $G$. Write $\overline{M}$ for $M/(G, \vartheta_G)$. There is a bidirectional correspondence between paths of $M$ and paths of $\overline{M}$. Formally we have the following:

1. If $x = s_0, s_1, s_2, \ldots$ is a path in $M$, then $\overline{x} = \overline{s_0}, \overline{s_1}, \overline{s_2}, \ldots$ is a path in $\overline{M}$ where $\overline{s_i} = \vartheta_G(s_i)$.

2. If $\overline{x} = \overline{s_0}, \overline{s_1}, \overline{s_2}, \ldots$ is a path in $\overline{M}$, then for every state $s'_0 \in M_S$ such that $\vartheta_G(s'_0) = \overline{s_0}$ there is a path $s'_0, s'_1, s'_2, \ldots$ such that $\vartheta_G(s'_i) = \overline{s_i}$.

**Proof.** 1. Given $(s_i, s_{i+1}) \in M_R$, then $(\overline{s}_i, \overline{s}_{i+1}) \in \overline{M}_R$.

2. Suppose that $(\overline{s}_i, \overline{s}_{i+1}) \in \overline{M}_R$. Then there is some $t, t' \in M_R$ with $t \in \overline{s}_i$ and $t' \in \overline{s}_{i+1}$ such that $s'_0 \in \overline{s}_i$, $s'_1 \in \overline{s}_1$, $s'_2 \in \overline{s}_2$ and $(s'_0, s'_1), (s'_1, s'_2), \ldots, (s'_{i-1}, s'_i) \in M_R$. Since both $t, s'_i \in \overline{s}_i$, there is a group element $g \in G$ such that $g(t) = s'_i$. Then $g(s'_0), g(s'_1), \ldots, g(s'_i) = t, s_{i+1}$ is a partial path in $M$ Continuing in this manner we see the existence of the desired path in $\overline{M}$.

We would like to extend the path correspondence between $M$ and $M/(G, \vartheta_G)$, for a group $G$ of state-mappings of $M$ and a representative map $\vartheta_G$ to one of substructures of Kripke structures. To do this, we need to identify a correspondence between $G$-closed substructures of $M$ and substructures of $M/(G, \vartheta_G)$. For $\mathcal{N}$ a $G$-closed substructure of $M$, there is a natural map $\mathcal{N} \rightarrow \mathcal{N}/(G, \vartheta_G)$.
Lemma 5.10. Let $\mathcal{M}$ be a Kripke structure and $G$ a group of state mappings of $\mathcal{M}$ with representative map $\vartheta_G$. Then for every substructure $\mathcal{N}'$ of $\mathcal{M}/(G, \vartheta_G)$, there is a $G$-closed substructure $\mathcal{N}''$ of $\mathcal{M}$ such that $\mathcal{N}'/(G, \vartheta_G) = \mathcal{N}''$.

Proof. We will construct a $G$-closed substructure $\mathcal{N}''$ of $\mathcal{M}$ such that $\mathcal{N}'/(G, \vartheta_G) = \mathcal{N}''$. For the states of $\mathcal{N}'$, we select exactly the $G$-orbits of the representative states of $\mathcal{N}'$, i.e., $\mathcal{N}'_S = \{g(s) : \forall g \in G, s \in \mathcal{N}_S\}$. For transitions, we note that for every transition $(s, t) \in \mathcal{N}_T'$ there is at least one transition $(s', t') \in \mathcal{M}_T$ such that $\vartheta(s') = s$ and $\vartheta(t') = t$. We add this transition and its orbit under $G$ to $\mathcal{N}'$. Hence for any state $s' \in \mathcal{N}'_S$ there is a transition $(s', t')$ for some $t' \in \mathcal{N}'_S$. We conclude that $\mathcal{N}''$ is total. Since $\mathcal{N}''$ is determined by its states and transitions as a substructure, the lemma is proved.

Lemma 5.10 establishes that the quotient map is surjective. The next lemma demonstrates that it is a homomorphism of the join-semilattices $\Lambda_{\mathcal{M}, G}$ and $\Lambda_{\mathcal{M}/(G, \vartheta_G)}$. We note that it is not a homomorphism of the lattices themselves, because the meet of two $G$-closed structures mapping might be empty due to the requirement that Kripke structures are total.

Lemma 5.11 (Quotient map respects join). Let $\mathcal{M}$ be a Kripke structure and $G$ a group of state mappings of $\mathcal{M}$ with representative map $\vartheta_G$. Let $\mathcal{N}, \mathcal{N}' \in \Lambda_{\mathcal{M}, G}$. Then

$$\mathcal{N} \lor \mathcal{N}' / (G, \vartheta_G) = \mathcal{N}' / (G, \vartheta_G) \lor \mathcal{N} / (G, \vartheta_G).$$

Proof. Clearly, by definition a state $\vartheta_G(s) \in (\mathcal{N} / (G, \vartheta_G) \lor \mathcal{N}' / (G, \vartheta_G))_S$ if and only if it is in at least one of $(\mathcal{N} / (G, \vartheta_G))_S$ or $(\mathcal{N}' / (G, \vartheta_G))_S$ which occurs if and only if $\vartheta_G(s)$ is in one of the $G$-closed substructures $\mathcal{N}$ or $\mathcal{N}'$.

For transitions, we note that a transition in $(\mathcal{N} / (G, \vartheta_G) \lor \mathcal{N}' / (G, \vartheta_G))_T$ is entirely contained in one of $(\mathcal{N} / (G, \vartheta_G))_T$ or $(\mathcal{N}' / (G, \vartheta_G))_T$. Hence the transition must come from either $\mathcal{N}$ or $\mathcal{N}'$.

As seen in Example 5.8 for a Kripke structure $\mathcal{M}$ with a group of state-mappings $G$ of $\mathcal{M}$ and representative map $\vartheta_G$, it is possible for multiple $G$-closed substructures of $\mathcal{M}$ to map to the same substructure of the quotient structure $\mathcal{M}/(G, \vartheta_G)$.

Recall that the join of $G$-closed substructures of $\mathcal{M}$ is $G$-closed.

Definition 5.12 ($G$-maximal). Let $\mathcal{M}$ be a Kripke structure with group $G$ of state-mappings and representative map $\vartheta_G$. A $G$-closed substructure $\mathcal{N}$ of $\mathcal{M}$ is $G$-maximal if

$$\mathcal{N} = \bigvee_{\mathcal{N}' \in \Lambda_{\mathcal{M}, G}} \mathcal{N}' / (G, \vartheta_G) \leq \mathcal{N} / (G, \vartheta_G).$$

Lemma 5.13. Let $\mathcal{M}$ be a Kripke structure and let $G$ be a group of state-mappings of $\mathcal{M}$ with representative map $\vartheta_G$. Let $\mathcal{M}', \mathcal{M}''$ be two $G$-maximal substructures of $\mathcal{M}$. Then $\mathcal{M}' \lor \mathcal{M}''$ is $G$-maximal.
Proof. Consider a $G$-closed substructure $N'$ such that $N'/\langle G, \vartheta_G \rangle \leq (M' \lor M'')/\langle G, \vartheta_G \rangle$. Let $(s, t) \in N'_T$. Since $(\vartheta_G(s), \vartheta_G(t)) \in (M' \lor M'')/\langle G, \vartheta_G \rangle$ we conclude that $(\vartheta_G(s), \vartheta_G(t))$ is in either $M'/\langle G, \vartheta_G \rangle$ or $M''/\langle G, \vartheta_G \rangle$. In either case, since $M'$ and $M''$ are both $G$-maximal the transitions $(g(s), g(t))$ are already contained in $M'$ or $M''$. 

**Lemma 5.14.** Let $M$ be a Kripke structure and let $G$ be a group of state-mappings of $M$ with representative map $\vartheta_G$. Let $M', M''$ be two $G$-maximal substructures of $M$. Then $M' \land M''$ is $G$-maximal.

**Proof.** Suppose that $N$ is the $G$-maximal substructure of $M$ corresponding to $(M' \land M'')/\langle G, \vartheta_G \rangle$. We note that $N/(G, \vartheta_G) \leq M'/\langle G, \vartheta_G \rangle \leq M''/(G, \vartheta_G)$. Hence $N \leq M'$. By symmetry $N \leq M''$ and thus $N \leq M' \land M''$. 

Combing Lemma 5.13 and 5.14 allows us to make the following definition.

**Definition 5.15** (G-maximal lattice of substructures). Let $M$ be a Kripke structure and let $G$ be a group of state-mappings of $M$ with representative map $\vartheta_G$. Then the set of $G$-maximal substructures of $M$ forms a sublattice $\Lambda_{M,G-\text{max}}$ of $\Lambda_M$.

While in general the quotient map from $\Lambda_M$ to $\Lambda_{M/\langle G, \vartheta_G \rangle}$ is always surjective, when restricted to $\Lambda_{M,G-\text{max}}$ the map is injective and a lattice isomorphism.

**Theorem 5.16** (G-Maximal Lattice Correspondence). Let $M$ be a Kripke structure and suppose that $G$ is a group of state-mappings of $M$ with a representative map $\vartheta_G$. Then the map $f(N) = N/\langle G, \vartheta_G \rangle$ is an isomorphism between the lattice of $G$-maximal substructures of $M$ and the lattice of structures of $M/\langle G, \vartheta_G \rangle$. That is $f$ is an isomorphism from $\Lambda_{M,G-\text{max}}$ to $\Lambda_{M/\langle G, \vartheta_G \rangle}$.

**Proof.** As noted the $G$-maximal lattice of substructures is in bijection with the lattice of substructures of $M/\langle G, \vartheta_G \rangle$. Since the meet and join of two $G$-maximal structures are determined entirely by the meet and join of their image in $M/\langle G, \vartheta_G \rangle$, the correspondence is a lattice isomorphism.

So far, our correspondence between models $M$ and $M/\langle G, \vartheta_G \rangle$ has been entirely syntactical. In the next section we explore the semantic connections between $M$ and $M/\langle G, \vartheta_G \rangle$.

### 5.3 Semantic relationships between structures and quotient structures.

Let $G$ be a symmetry group of $M$.

**Definition 5.17.** A CTL formula $f$ is $G$-invariant over $M$, if for every state $s$, every $g \in G$, for all maximal propositional subformulae $P$ of $f$, we have $M, s \models P \iff M, g(s) \models P$.

**Lemma 5.18.** Let $G$ be a group of state-mappings for a Kripke structure $M$. If $f$ is $G$-invariant, then the valuation of $f$ in $\overline{M} = M/\langle G, \vartheta_G \rangle$ does not depend on the choice of representative map $\vartheta_G$. 

11
Proof. Let $\vartheta_G, \varphi_G$ be two representative maps with respect to $G$. The quotients $\mathcal{M}/(G, \vartheta_G)$ and $\mathcal{M}/(G, \varphi_G)$ have corresponding transition maps. Moreover, for any maximal propositional subformula, $P$ of $f$ we have that

$$\mathcal{M}/(G, \vartheta_G), \vartheta_G(s) \models P \iff \mathcal{M}/(G, \varphi_G), \varphi_G(s) \models P.$$  

The valuation of $f$ depends only on the valuation of its maximal propositional subformulae at all other states of the model. Since $\vartheta$ and $\varphi$ have the same evaluation of the maximal propositional subformulae at all states, the evaluation of $f$ at all states is the same.

This allows us to connect semantic statements about $\mathcal{M}$ with semantic statements about $\mathcal{M}/(G, \vartheta_G)$ for formulae that are $G$-invariant.

If $G$ is a group of state-mappings of a structure $\mathcal{M}$ and $\vartheta_G$ is a representative map of $G$, then the path correspondence theorem establishes what we call a $G$-bisimulation between $\mathcal{M}$ and the structure $\mathcal{M}/(G, \vartheta_G)$.

**Definition 5.19 (G-bisimulation).** Let $\mathcal{M}$ be a Kripke structure, $G$ be a group of state-mappings of $\mathcal{M}$ with representative map $\vartheta_G$, and $\overline{\mathcal{M}} = \mathcal{M}/(G, \vartheta_G)$. Let $R \subseteq \mathcal{M}_S \times \overline{\mathcal{M}}_S$. Then $R$ is a $G$-bisimulation between $\mathcal{M}$ and $\overline{\mathcal{M}}$ iff:

1. If $(s, t) \in R$ then there exists $g \in G$ such that $t = g(s)$, i.e., $t = \vartheta_G(s)$;
2. If $(s, t) \in R$ and $s \rightarrow s' \in \mathcal{M}_T$, then there exists $t'$ such that $t \rightarrow t' \in \overline{\mathcal{M}}_T$ and $(s', t') \in R$
3. If $(s, t) \in R$ and $t \rightarrow t' \in \overline{\mathcal{M}}_T$, then there exists $s'$ such that $s \rightarrow s' \in \mathcal{M}_T$ and $(s', t') \in R$

**Lemma 5.20.** Let $\mathcal{M}$ be a Kripke structure, $G$ be a group of state-mappings of $\mathcal{M}$ with representative map $\vartheta_G$, and $\overline{\mathcal{M}} = \mathcal{M}/(G, \vartheta_G)$. Let $\sim = (s, \overline{s})$ where $s \in \mathcal{M}_S$ and $\overline{s} = \vartheta_G(s)$. Then $\sim$ is a $G$-bisimulation.

Proof. We show that all clauses of Definition 5.19 hold. By definition of $\vartheta_G, \overline{s} = g(s)$ for some $g \in G$, so Clause 1 holds. Theorem 5.9 shows the existence of the corresponding transitions needed to satisfy Clauses 2 and 3 of Definition 5.19.

**Lemma 5.21.** Let $\mathcal{M}$ be a Kripke structure, $G$ be a group of state-mappings of $\mathcal{M}$ with representative map $\vartheta_G$, and $\overline{\mathcal{M}} = \mathcal{M}/(G, \vartheta_G)$. Let $s \in \mathcal{M}_S$, $t \in \overline{\mathcal{M}}_S$, let $\sim = (s, \overline{s}$ where $s \in \mathcal{M}_S$ and $\overline{s} = \vartheta_G(s)$. If $s \sim t$. Let $f$ be a $G$-invariant CTL formula. Then $\mathcal{M}, s \models f \iff \overline{\mathcal{M}}, t \models f$.

Proof. Proof by induction on the structure of CTL formulae, as given by the grammar in Section 3.

Since $f$ is $G$-invariant over $\mathcal{M}$, then for all $t \in \mathcal{M}$ we have that $\mathcal{M}, t \models P$ if $\overline{\mathcal{M}}, \vartheta_G(t) \models P$, where $P$ is any maximal propositional subformula of $f$.

This provides the base case for the induction, where $f$ is an atomic proposition.

The induction step when $f$ has main operator one of $\neg, \land, \lor$ are straightforward, and omitted. The induction steps when $f$ is one of $\AX g$, $\EX g$ and $\AG \Box g \land h$ use Definition 5.19 to establish the existence of the necessary transitions/paths. The details are standard, see for example Theorem 3.2 in [4].

12
Lemma 5.22. Let $M$ be a Kripke structure, $G$ be a group of state-mappings of $M$ with representative map $\vartheta_G$, and $\overline{M}$ be $M/(G, \vartheta_G)$. Let $f$ be a $G$-invariant CTL formula and $s \in M_S$. Then $M, s \models f$ iff $\overline{M}, \vartheta_G(s) \models f$.

Proof. By Lemma 5.20 $s \sim \vartheta_G(s)$. So, by Lemma 5.21 $M, s \models f$ iff $\overline{M}, \vartheta_G(s) \models f$. □

Definition 5.23 (Repair). Given a structure $M$ and a formula $f$, we call a nonempty substructure $M'$ of $M$ a repair of $M$ with respect to $f$ if $M' \models f$.

If a CTL formula $f$ is $G$-invariant, then the lattice correspondence will respect the valuation of $f$.

Theorem 5.24 (Repair Correspondence). Let $M$ be a Kripke structure, $G$ be a group of state-mappings of $M$ with a representative map $\vartheta_G$, and $f$ be a $G$-invariant CTL formula.

Let $N$ be a $G$-closed substructure of $M$, and let $\overline{N} = N/(G, \vartheta_G)$. Then $N, s \models f$ iff $\overline{N}, \vartheta_G(s) \models f$.

Proof. We can show that $N$ and $\overline{N}$ are $G$-bisimilar under the usual relation of each state $s$ in $N_S$ with $\vartheta_G(s)$ in $\overline{N}_S$. □

It should be noted however that there are formulae $f$ and structures $M$, such that $f$ is $G$-invariant for some symmetry group $G$ of $M$ and there are no $G$-closed repairs of $f$, but there are repairs that are not $G$-closed. That is, not every repairable structure has a symmetric repair.

Theorem 5.24 gives us the commuting diagram in Figure 4.

6 Repair of concurrent programs

We now apply the repair correspondence theorem to realize our goal: the repair of concurrent programs. A concurrent program $P = P_1 \parallel \ldots \parallel P_n$ consists of $n$ sequential processes executing in parallel. Each process $P_i$ is a set of $i$-actions, i.e., tuples $(s_i, B \rightarrow A, t_i)$, where $s_i, t_i$ are local states of $P_i$, $B$ is a guard, and $A$ is an assignment to shared variables. We say action when we wish to ignore the identity of the process that executes the action. Each $P_i$ has a local set of atomic propositions $AP_i$ which it alone can modify, but other processes can read. Each local state of $P_i$ determines as assignment of truth values to the atomic propositions in $AP_i$. We set $AP = AP_1 \cup \ldots \cup AP_n$. A global state

\[ \begin{align*}
M & \xrightarrow{/(G, \vartheta_G)} \overline{M} \\
\vdash & \udownarrow \vdash \\
M' \models f & \xrightarrow{/(G, \vartheta_G)} \overline{N}' \models f \\
G \text{ closed} & \end{align*} \]

Figure 4: Commutative diagram illustrating the connection between a structure $M$, a quotient structure $\overline{M}$, and substructures of $\overline{M}$ satisfying a formula $f$ with $G$-closed substructures of $M$ satisfying $f$. 
$s = (s_1, \ldots, s_n, v_1, \ldots, v_m)$ of $P$ is a tuple consisting of local states $s_1, \ldots, s_n$ for processes $P_1, \ldots, P_n$ respectively, and values $v_1, \ldots, v_m$ for shared variables $x_1, \ldots, x_m$. Define $s \upharpoonright i = s_i$ for $i = 1, \ldots, n$ and $s(x_j) = v_j$ for $j = 1, \ldots, m$. The truth values that atomic propositions in $AP_i$ have in $s$ must be the same as those in $s_i$. The states of Kripke structures that we have treated above are global states as described here.

Given a concurrent program $P$ and a CTL formula $f$, we wish to modify $P$ to produce a $P'$ such that $M \models f$, where $M$ is the global state transition graph of $P'$. The modification is "subtractive", that is, it only removes behaviors, it does not add behaviors (which would be, in essence, a synthesis, and not a repair, algorithm). We proceed as follows:

1. Generate the symmetry-reduced state transition graph $N$ of $P$. We use the algorithm given by Figure 1 in [26], which is reproduced in Appendix ?.

2. Repair $N$ w.r.t. the given CTL specification $f$. Let $N'$ be the repaired structure, so that $N' \models f$.

3. Extract a repaired concurrent program $P'$ from $N'$

Steps 2 and 3 are given in the subsections below.

6.1 Repair of symmetry-reduced structures

We apply the model repair algorithm of [2] to the symmetry-reduced state transition graph $N$ of $P$, and the specification $f$ of $P$. This algorithm is sound and complete, so that if $N$ has some substructure that satisfies $f$, then the algorithm will return such a substructure $N'$. If not, the algorithm will report that no repair exists, in which case there is no subtractive modification of $P$ which satisfies $f$.

6.2 Extraction of concurrent programs from symmetry-reduced structures

We extract concurrent program $P'$ from $N'$ following [15], by projecting each transition in $s \overset{a}{\rightarrow} t \in N_T$ onto the process $P_i$ making the transition. For each $s \overset{a}{\rightarrow} t \in N_T$ we define $\text{action}(s \overset{a}{\rightarrow} t)$ to be the $i$-action $(s \upharpoonright i, B, A, t \upharpoonright i)$ as follows. The start, end states of this action are the projections of $s,t$ respectively onto $P_i$. We define the guard $B = \langle \bigwedge_{Q \in \mathcal{N}_L(s)} Q \rangle \land \langle \bigwedge_{Q \notin \mathcal{N}_L(s)} \neg Q \rangle \land \langle \bigwedge_{x \in \mathcal{SH}} x = s(x) \rangle$ where $Q$ ranges over $AP - AP_i$. When $P_i$ is in local state $s_i$, the guard $B$ checks that the current global state is actually $s$. We define the assignment $A = \langle /x := t(x) \rangle$ where $x$ ranges over all shared variables such that $t(x) \neq s(x)$. That is, $A$ makes the necessary updates to the shared variables in moving from state $s$ to state $t$.

Let $AC = \{ \text{action}(s \overset{a}{\rightarrow} t) \mid s \overset{a}{\rightarrow} t \in N_T \}$, i.e., $AC$ is the set of actions generated by all the transitions of $N$. In principle, each process $P'_i$ of the repaired program $P'$ can be set to be the set of $i$-actions in $AC$. It is straightforward to show that $P'$ thus constituted generates exactly $N'$ as its global state transition diagram. Since $N' \models f$, we also have $P' \models f$ as desired. However, even
though $N'$ is total, it is possible for some processes in $P'$ to have "dead end" local states from which they have no outgoing actions. We therefore repeatedly apply some $g \in G$ to $AC$, until all such "dead end" local states acquire at least one outgoing action. One can continue this application of $g \in G$ to $AC$ until no more actions are generated, which produces a "G-closed" program $P'$. This is the "largest" correct program that can be produced as a repair of $P$. Using the largest program may increase the available concurrency, but may produce a large number of actions, and so increase the complexity of the repair procedure.

**Theorem 6.1.** Let $N'$ be a repaired Kripke structure as above, $P'$ be a concurrent program extracted from $N'$ as above, and $M'$ be the global state transition diagram of $P'$. Then every initial state of $M'$ is $G$-bisimilar to some initial state of $N'$, and vice-versa.

Proof. Let $\sim$ consist of the pairs $(s, t)$ such that $s \in N'$, $t \in M'$, and $t = g(s)$ for some $g \in G$. The proof that $\sim$ is a $G$-bisimulation follows from the definitions of program extraction and program execution semantics. By the definitions of program extraction and program execution semantics, $N'$ and $M'$ have isomorphic sets of initial states, with corresponding states having the same labels. It follows that these states are bisimilar. Hence $\sim$ is a $G$-bisimulation which moreover relates each initial state of $N'$ to an initial state of $M'$ and vice-versa. 

**Lemma 6.2.** Let $N'$ be a Kripke structure as above that is repaired w.r.t. a $G$-invariant CTL formula $f$. Let $P'$ be a concurrent program extracted from $N'$ as above, and $M'$ be the global state transition diagram of $P'$. Then $M' \models f$.

Proof. Since $N'$ is repaired w.r.t. $f$, we have $N' \models f$ by Corollary 4.5 of [2]. Hence $N', s_0 \models f$ for every initial state $s_0$ of $N'$. By Theorem 6.1 every initial state of $M'$ is $G$-bisimilar to some initial state of $N'$. Hence by Lemma 5.21 $M', t_0 \models f$ for every initial state $t_0$ of $M'$. Hence $M' \models f$. 

7 Examples

7.1 Two process mutual exclusion

Figure 5 shows a Kripke structure $M$ for mutual exclusion of two processes $P_1$ and $P_2$. $P_i$ ($i = 1, 2$) can be in any of three local states: $Ni$ (neutral, doing local computation), $Ti$ (trying, has requested critical section entry), and $Ci$ (in the critical section). $M$ has exactly two symmetries: the identity map, and the map that swaps process indices 1 and 2. We construct the quotient model under this symmetry group. The reduced structure $\overline{M} = M/(G, \partial_G)$ is also shown in Figure 5. Transitions of $P_1$, $P_2$ are shown in blue, red, respectively. $\overline{M}$ has a transition (shown in black) from state $S6$ to $S1$, which is the quotient of the transition from $S6$ to $S2$ in $M$, i.e., $\partial_G(S6) = S6$ and $\partial_G(S2) = S1$ so the edge $(\partial_G(S6), \partial_G(S1))$ occurs in $\overline{M}$.

Consider the formula $\varphi = \text{AG} \neg (C_1 \land C_2)$. Clearly, $M \not\models \varphi$. Since $\varphi$ is $G$-invariant, this is equivalent to $\overline{M} \not\models \varphi$ for any representative map $\partial_G$. From the correspondence theorem, any substructure of $\overline{M}$ that satisfies $\varphi$ corresponds to a $G$-maximal substructure of $M$ that satisfies $\varphi$.

Figure 6 shows the repair of the reduced structure $\overline{M}$, and the resultant lifting of the repair to $M$. The deleted transitions and states are shown dashed.
Figure 5: The original model $\mathcal{M}$ and quotient $\overline{\mathcal{M}} = \mathcal{M}/(G, \vartheta_G)$. 
Figure 6: The repair of $\overline{\mathcal{M}}$ and the lifting of the repair to $\mathcal{M}$.
7.2 N-Process Mutual Exclusion

We now consider mutual exclusion for \(N\)-processes. To reduce clutter, we remove the trying \((T_i)\) state, and we give a concrete example for 3 processes — the generalization to \(N\) processes is straightforward. We start with the naïve concurrent program shown in Figure 7, where actions are labeled with the trivial guard \(true\). The resulting Kripke structure is an \(N\)-dimensional hypercube, and Figure 8 shows the structure \(M\) for 3-processes, along with the quotient \(M = M/(G, \vartheta_G)\). We consider the mutual exclusion specification \(\bigwedge_{i \neq j} AG \neg (C_i \land C_j)\). The symmetry group \(G\) for both structure and specification is the full permutation group on the indices \(\{1, \ldots, N\}\). For \(N\)-processes, we have that the quotient model by the full group of symmetries has \(N + 1\) states, while the original model would have \(2^N\) states. Figure 9 shows the repair of the quotient \(\overline{M}\) and then the lifting of the repair to the original structure \(M\). Figure 10 shows the correct (repaired) program \(P'\) that is extracted from the repaired quotient in Figure 9. For \(N\) processes, the guard on actions of \(P'_i\) is \(\bigwedge_{j \neq i} N_j\).

8 Conclusions

We presented a theory for the substructures of a given Kripke structure \(M\), their organization into lattices, and how these substructures can interact with a group of state-mappings of \(M\). We showed the existence of an isomorphism between the lattice \(\Lambda_{M, G-\text{max}}\) of \(G\)-maximal substructures of \(M\) and the lattice of substructures of \(M(G, \vartheta_G)\). Using this theory we were able to devise a method for repairing concurrent programs which exploits symmetry.
Figure 8: The original model $\mathcal{M}$ and quotient $\overline{\mathcal{M}} = \mathcal{M}/\langle G, \vartheta_G \rangle$. 
Figure 9: The repair of $\overline{M}$ and the lifting of the repair to $M$.

Figure 10: The repaired program $P'$
References

[1] Benjamin Aminof, Swen Jacobs, Ayrat Khalimov, and Sasha Rubin. Parameterized model checking of token-passing systems. In Kenneth L. McMillan and Xavier Rival, editors, Verification, Model Checking, and Abstract Interpretation - 15th International Conference, VMCAI 2014, San Diego, CA, USA, January 19-21, 2014, Proceedings, volume 8318 of Lecture Notes in Computer Science, pages 262–281. Springer, 2014. doi:10.1007/978-3-642-54013-4_15

[2] Paul C. Attie, Kinan Dak-Al-Bab, and Mouhammad Sakr. Model and program repair via SAT solving. ACM Trans. Embed. Comput. Syst., 17(2):32:1–32:25, 2018. doi:10.1145/3147426

[3] Roderick Bloem, Swen Jacobs, Ayrat Khalimov, Igor Konnov, Sasha Rubin, Helmut Veith, and Josef Widder. Decidability of Parameterized Verification. Synthesis Lectures on Distributed Computing Theory. Morgan & Claypool Publishers, 2015. doi:10.2200/S00658ED1V01Y201508DCT013

[4] Michael C. Browne, Edmund M. Clarke, and Orna Grumberg. Characterizing finite kripke structures in propositional temporal logic. Theoretical computer science, 59(1-2):115–131, 1988.

[5] F. Buccafurri, T. Eiter, G. Gottlob, and N. Leone. Enhancing model checking in verification by AI techniques. Artif. Intell., 112:57–104, 1999.

[6] M. Carrillo and D.A. Rosenblueth. A method for ctl model update, representing kripke structures as table systems. IJPAM, 52:401–431, 2009.

[7] G. Chatzieleftheriou, B. Bonakdarpour, S.A. Smolka, and P. Katsaros. Abstract model repair. In AlwynE. Goodloe and Suzette Person, editors, NASA Formal Methods, volume 7226 of Lecture Notes in Computer Science, pages 341–355. Springer Berlin Heidelberg, Norfolk, VA, USA, 2012.

[8] Edmund M. Clarke, E. Allen Emerson, Somesh Jha, and A. Prasad Sistla. Symmetry reductions in model checking. In CAV, volume 1427 of Lecture Notes in Computer Science, pages 147–158. Springer, 1998.

[9] Edmund M. Clarke, E. Allen Emerson, and A. Prasad Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. ACM Trans. Program. Lang. Syst., 8(2):244–263, 1986.

[10] Edmund M. Clarke, Orna Grumberg, and Somesh Jha. Verifying parameterized networks. ACM Trans. Program. Lang. Syst., 19(5):726–750, 1997. doi:10.1145/265943.265960

[11] Edmund M. Clarke, Somesh Jha, Reinhard Enders, and Thomas Filkorn. Exploiting symmetry in temporal logic model checking. Formal Methods Syst. Des., 9(1/2):77–104, 1996. doi:10.1007/BF00625969

[12] Edmund M. Clarke, Muralidhar Talupur, Tayssir Touili, and Helmut Veith. Verification by network decomposition. In Philippa Gardner and Nobuko Yoshida, editors, CONCUR 2004 - Concurrency Theory, 15th International
[13] Edmund M. Clarke, Muralidhar Talupur, and Helmut Veith. Environment abstraction for parameterized verification. In E. Allen Emerson and Kedar S. Namjoshi, editors, Verification, Model Checking, and Abstract Interpretation, 7th International Conference, VMCAI 2006, Charleston, SC, USA, January 8-10, 2006, Proceedings, volume 3855 of Lecture Notes in Computer Science, pages 126–141. Springer, 2006. doi:10.1007/11609773_9

[14] E. Allen Emerson. Temporal and modal logic. In Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics (B), pages 995–1072. Elsevier and MIT Press, 1990.

[15] E. Allen Emerson and Edmund M. Clarke. Using branching time temporal logic to synthesize synchronization skeletons. Sci. Comput. Program., 2(3):241–266, 1982.

[16] E. Allen Emerson, John Havlicek, and Richard J. Trefler. Virtual symmetry reduction. In LICS, pages 121–131. IEEE Computer Society, 2000.

[17] E. Allen Emerson and Vineet Kahlon. Reducing model checking of the many to the few. In CADE, volume 1831 of Lecture Notes in Computer Science, pages 236–254. Springer, 2000.

[18] E. Allen Emerson and Vineet Kahlon. Model checking large-scale and parameterized resource allocation systems. In TACAS, volume 2280 of Lecture Notes in Computer Science, pages 251–265. Springer, 2002.

[19] E. Allen Emerson and Vineet Kahlon. Exact and efficient verification of parameterized cache coherence protocols. In CHARME, volume 2860 of Lecture Notes in Computer Science, pages 247–262. Springer, 2003.

[20] E. Allen Emerson and Vineet Kahlon. Model checking guarded protocols. In LICS, pages 361–370. IEEE Computer Society, 2003.

[21] E. Allen Emerson and Vineet Kahlon. Rapid parameterized model checking of snoopy cache coherence protocols. In TACAS, volume 2619 of Lecture Notes in Computer Science, pages 144–159. Springer, 2003.

[22] E. Allen Emerson and Kedar S. Namjoshi. Reasoning about rings. In POPL, pages 85–94. ACM Press, 1995.

[23] E. Allen Emerson and Kedar S. Namjoshi. Automatic verification of parameterized synchronous systems (extended abstract). In CAV, volume 1102 of Lecture Notes in Computer Science, pages 87–98. Springer, 1996.

[24] E. Allen Emerson and Kedar S. Namjoshi. Verification of parameterized bus arbitration protocol. In CAV, volume 1427 of Lecture Notes in Computer Science, pages 452–463. Springer, 1998.

[25] E. Allen Emerson and Kedar S. Namjoshi. On reasoning about rings. Int. J. Found. Comput. Sci., 14(4):527–550, 2003.
[26] E. Allen Emerson and A. Prasad Sistla. Symmetry and model checking. *Formal methods in system design*, 9(1):105–131, 1996.

[27] E. Allen Emerson and A. Prasad Sistla. Utilizing symmetry when model-checking under fairness assumptions: An automata-theoretic approach. *ACM Trans. Program. Lang. Syst.*, 19(4):617–638, 1997.

[28] E. Allen Emerson and Richard J. Trefler. Model checking real-time properties of symmetric systems. In *MFCS*, volume 1450 of *Lecture Notes in Computer Science*, pages 427–436. Springer, 1998.

[29] E. Allen Emerson and Richard J. Trefler. From asymmetry to full symmetry: New techniques for symmetry reduction in model checking. In *CHARME*, volume 1703 of *Lecture Notes in Computer Science*, pages 142–156. Springer, 1999.

[30] E. Allen Emerson and Thomas Wahl. Dynamic symmetry reduction. In *TACAS*, volume 3440 of *Lecture Notes in Computer Science*, pages 382–396. Springer, 2005.

[31] Javier Esparza, Alain Finkel, and Richard Mayr. On the verification of broadcast protocols. In *14th Annual IEEE Symposium on Logic in Computer Science*, Trento, Italy, July 2-5, 1999, pages 352–359. IEEE Computer Society, 1999. doi:10.1109/LICS.1999.782630.

[32] Steven M. German and A. Prasad Sistla. Reasoning about systems with many processes. *J. ACM*, 39(3):675–735, 1992. doi:10.1145/146637.146681.

[33] Nouraldin Jaber, Swen Jacobs, Christopher Wagner, Milind Kulkarni, and Roopsha Samanta. Parameterized verification of systems with global synchronization and guards. In Shuvendu K. Lahiri and Chao Wang, editors, *Computer Aided Verification - 32nd International Conference, CAV 2020, Los Angeles, CA, USA, July 21-24, 2020, Proceedings, Part I*, volume 12224 of *Lecture Notes in Computer Science*, pages 299–323. Springer, 2020. doi:10.1007/978-3-030-53288-8_15.

[34] B. Jobstmann, A. Griesmayer, and R. Bloem. Program repair as a game. In *CAV*, pages 226–238, Berlin, Heidelberg, 2005. Springer-Verlag.

[35] S. Staber, B. Jobstmann, and R. Bloem. Finding and fixing faults. In *CHARME ’05*, pages 35–49, Berlin, Heidelberg, 2005. Springer-Verlag. Springer LNCS no. 3725.

[36] Christian von Essen and Barbara Jobstmann. Program repair without regret. *Formal Methods Syst. Des.*, 47(1):26–50, 2015. doi:10.1007/s10703-015-0223-6.
A CTL Semantics

Definition A.1. \( M, s \models \varphi \) means that formula \( \varphi \) is true in state \( s \) of structure \( M \), and \( M, s \not\models \varphi \) means that \( \varphi \) is false in \( s \).

1. \( M, s \models true \) and \( M, s \not\models false \)
2. \( M, s \models p \iff p \in L(s) \) where atomic proposition \( p \in AP \)
3. \( M, s \models \neg \varphi \iff M, s \not\models \varphi \)
4. \( M, s \models \varphi \land \psi \iff M, s \models \varphi \) and \( M, s \models \psi \)
5. \( M, s \models \varphi \lor \psi \iff M, s \models \varphi \) or \( M, s \models \psi \)
6. \( M, s \models AX \varphi \iff \) for all \( t \) such that \((s, t) \in T : (M, t) \models \varphi \)
7. \( M, s \models EX \varphi \iff \) there exists \( t \) such that \((s, t) \in T \) and \((M, t) \models \varphi \)
8. \( M, s \models A[\varphi \mathcal{R}\psi] \iff \) for all fullpaths \( \pi = s_0, s_1, \ldots \) starting from \( s = s_0 : \forall k \geq 0 : (\forall j < k : M, s_j \not\models \varphi) \) implies \( M, s_k \models \psi \)
9. \( M, s \models E[\varphi \mathcal{R}\psi] \iff \) for some fullpath \( \pi = s_0, s_1, \ldots \) starting from \( s = s_0 : \forall k \geq 0 : (\forall j < k : M, s_j \not\models \varphi) \) implies \( M, s_k \models \psi \)