Working Paper Series
ISSN 1177-777X

PARTIAL UNFOLDING FOR COMPOSITIONAL NONBLOCKING VERIFICATION OF EXTENDED FINITE-STATE MACHINES

Sahar Mohajerani, Robi Malik, Martin Fabian

Working Paper: 01/2013
January 30, 2013

©Sahar Mohajerani, Robi Malik, Martin Fabian

Department of Computer Science
The University of Waikato
Private Bag 3105
Hamilton, 3240
New Zealand

PARTIAL UNFOLDING FOR COMPOSITIONAL NONBLOCKING VERIFICATION OF EXTENDED FINITE-STATE MACHINES

Sahar Mohajerani
Department of Signals and Systems
Chalmers University of Technology
Göteborg, Sweden
mohajera@chalmers.se

Robi Malik
Department of Computer Science
The University of Waikato
Hamilton, New Zealand
robi@waikato.ac.nz

Martin Fabian
Department of Signals and Systems
Chalmers University of Technology
Göteborg, Sweden
fabian@chalmers.se

January 30, 2013

Abstract

This working paper describes a framework for *compositional nonblocking verification* of reactive systems modelled as *extended finite-state machines*. The *nonblocking* property can capture the absence of livelocks and deadlocks in concurrent systems. Compositional verification is shown in previous work to be effective to verify this property for large *discrete event systems*. Here, these results are applied to extended finite-state machines communicating via shared memory. The model to be verified is composed gradually, simplifying components through *abstraction* at each step, while *conflict equivalence* guarantees that the final verification result is the same as it would have been for the non-abstracted model. The working paper concludes with an example showing the potential of compositional verification to achieve substantial state-space reduction.

1 Introduction

Reactive systems are typically safety-critical, where failures can result in huge financial losses, or even human fatalities. Thus, logical correctness is a crucial property of most reactive systems, and formal verification is an important part of guaranteeing logical correctness. In the field of *model* *checking* [3], various methods have been developed to verify reactive systems of increasing size and complexity, most notably *symbolic model checking* [17] and *abstraction* [7].

Formal verification requires a formal model, and *finite-state machines (FSM)* [12] are widely used in the literature to represent reactive systems. FSMs describe the dynamic behaviour of a reactive system by *states*, where certain conditions hold, and *transitions* between these states that change the conditions. For systems with data dependency, it is natural to extend FSMs with variables that represent data. This results in *extended finite-state machines (EFSM)*, which have been similarly defined by several researchers [5, 6, 21, 23].

An important aspect of correctness is the absence of *livelocks* and *deadlocks*. FSMs (and EFSMs) allow certain states to be designated as *terminal* states. The *nonblocking* property [20] requires that the system should from any reachable state always be able to reach some terminal state. This property is used in supervisory control theory of discrete events systems [20] to capture the absence of livelocks and deadlocks.

Expressed in CTL [3], nonblocking can be written as $\mathbf{AG}\,\mathbf{EF}\ \ terminal\_state$. In [7], for the purpose of abstraction in model checking, $\forall$CTL* is defined as a subset of CTL where only universal path quantification is allowed. If a given $\forall$CTL* property is satisfied by all components of a system, the property is also satisfied by the composed system. However, nonblocking cannot be expressed in $\forall$CTL*, which makes it impossible to use many standard abstraction techniques for nonblocking verification.

Compositional methods [9] exploit the compositional structure of a system, i.e., the fact that the system is made up of several FSMs interacting with each other. Abstraction is used to remove states and transitions that are superfluous for the purpose of verification of the property at hand. While compositional methods have shown impressive results for FSMs [9, 19], their adaptation to EFSMs is still in its infancy. Transforming an EFSM to a FSM [14, 21] makes it possible to apply the algorithms for FSMs to an EFSM model. However, the transformation has the drawback of significantly increasing the number of transitions in the system, or losing the compositional structure.

This working paper generalises the compositional verification method [9] to be applicable directly to reactive systems modelled as EFSMs. *Partial unfolding* is proposed to remove a variable from the system, and *symbolic observation equivalence* is introduced to be applied to EFSMs directly without the need for transforming EFSMs to FSMs. Furthermore, another abstraction method, called the *Active Events Rule* [9], is extended in the framework of EFSMs, and has great potential to abstract systems while preserving the nonblocking property.

The remainder of the working paper is structured as follows. Sect. 2 introduces extended finite state machines, and section 3 gives an example of a concurrent program modelled by EFSMs. Next, section 4 describes the process of converting EFSMs to FSMs, and section 5 presents some experiments with FSM-based compositional verification applied to the example from section 3. Then section 6 presents different ways of computing abstractions that can be applied directly on EFSMs, and section 7 demonstrates compositional abstraction-based verification on EFSMs, using the same example. Finally, section 8 adds some concluding remarks. Formal proofs of all technical results are in the appendix.

2 Extended Finite-State Machines

In this working paper, reactive systems are modelled as *extended finite-state machines (EFSM)* that synchronise in interleaving semantics and communicate via shared memory. Extended finite-state machines are similar to conventional finite-state machines (FSM) [12], but augmented with *updates* associated to the transitions [5, 6, 21]. Updates are formulas over bounded discrete variables.

A *variable* v is an entity associated with a finite *domain* $\mathrm{dom}(v)$ and an *initial value* $v^\circ \in \mathrm{dom}(v)$. A second set of variables, called *next-state variables* and denoted by $V' = \{\, v' \mid v \in V \,\}$ with $\mathrm{dom}(v') = \mathrm{dom}(v)$, is used to describe how variables are updated.

An *update* is a formula using variables from $V \cup V'$. For example, let x be a variable with domain $\mathrm{dom}(x) = \{0, \dots, 5\}$. An update $x' = x + 1$ changes the variable x by adding 1 to its current value, if it currently is less than 5. Otherwise (if $x = 5$) the transition is disabled and no updates are performed. Another possibility is to write the formula $x' = \min(5, x + 1)$, in which case the transition remains enabled when $x = 5$. The update $x = 3$ disables a transition unless $x = 3$ in the current state, and leaves the value of x in the next state, x', unchanged. Differently, the update $x' = 3$ is always enabled, and the value of x in the next state is forced to be 3. The set of all update formulas using variables in V or V' is denoted by Π_V.

Definition 1 An *Extended Finite-State Machine (EFSM)* is a tuple $E = \langle V, Q, \rightarrow, Q^\circ, Q^\omega \rangle$, where V is a finite set of variables, Q is a finite set of *locations*, $\rightarrow \subseteq Q \times \Pi_V \times Q$ is the *conditional transition relation*, $Q^\circ \subseteq Q$ is the set of *initial locations*, and $Q^\omega \subseteq Q$ is the set of *terminal locations*.

The expression $x \xrightarrow{p} y$ denotes the presence of a transition in E, from location x to location y with update $p \in \Pi_V$. On the occurrence of such a transition, the EFSM changes its location from x to y while updating its variables in accordance with p; variables that do not occur as next-state variables in p remain unchanged.

Usually, reactive systems are modelled as several components interacting with each other. An *EFSM system* is a collection of interacting EFSMs,

$$\mathcal{E} = \{E_1, \dots, E_n\} \ . \tag{1}$$

The behaviour of such a system is expressed using *interleaving* semantics [3].

Definition 2 Given two EFSMs $E = \langle V_E, Q_E, \rightarrow_E, Q_E^\circ, Q_E^\omega \rangle$ and $F = \langle V_F, Q_F, \rightarrow_F, Q_F^\circ, Q_F^\omega \rangle$ the *composition* of E and F is

$$E \parallel F = \langle V_E \cup V_F, Q_E \times Q_F, \rightarrow, Q_E^\circ \times Q_F^\circ, Q_E^\omega \times Q_F^\omega \rangle \ , \tag{2}$$

where

- $(x_E, x_F) \xrightarrow{p_E} (y_E, x_F)$ if $x_E \xrightarrow{p_E}_E y_E$;
- $(x_E, x_F) \xrightarrow{p_F} (x_E, y_F)$ if $x_F \xrightarrow{p_F}_F y_F$.

```
                do
                    $s_0$ : $x_0 =$ random number;
source:             $s_1$ : **await** $c_0 = 0$;
                    $s_2$ : $c_0 = x_0$;
                loop
    ||
                do
                    $q_0$ : **await** $c_{k-1} > 0$;
                    $q_1$ : $x_k = c_{k-1}$;
                    $q_2$ : $c_{k-1} = 0$;
sieve$_k$:          $q_3$ : **if** $x_k = p_k \vee x_k \bmod p_k \neq 0$ **then**
                                $q_5$ : **await** $c_k = 0$;
                                $q_6$ : $c_k = x_k$;
                            **end**
                loop
    ||
                do
                    $r_0$ : **await** $c_n > 0$;
sink:               $r_1$ : print $c_n$;
                            $c_n = 0$;
                loop
```

Algorithm 1: Distributed Sieve of Eratosthenes.

3 Example

This section shows how a concurrent program can be modelled using EFSMs. The same example is used throughout the working paper to explain different approaches to compositional nonblocking verification.

Algorithm 1 shows a distributed version of the *Sieve of Eratosthenes* for generating prime numbers. The system consists of two processes source and sink, plus a variable number of sieve processes sieve$_k$. The source generates numbers x_0 from a finite set (program location s_0) and sends them to the first sieve process sieve$_1$ using the shared variable c_0 (s_1 and s_2). There are n sieve processes for the first n prime numbers $p_1, \ldots, p_n$. The k-th sieve process sieve$_k$, upon receiving a new number x_k through c_{k-1} (q_0 and q_1), tests whether the number is equal to or divisible by its prime number p_k (q_3). If the received number is different from and divisible by p_k, it is discarded, otherwise it is sent to the next sieve process sieve$_{k+1}$ using the shared variable c_k (q_5 and q_6). Numbers that pass through n sieve processes are received by the sink (r_0), which prints them before releasing the shared variable c_n (r_1).

Figure 1 shows an EFSM model of this system. Initial locations are marked with an incoming

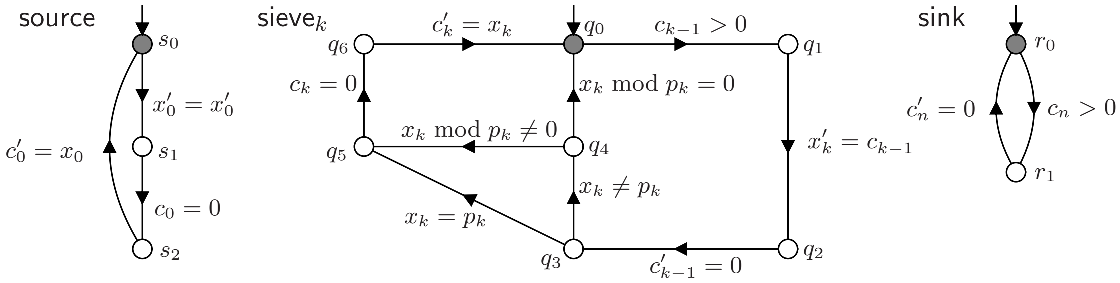


Figure 1: EFSM model of Distributed Sieve of Eratosthenes.

arrow, and terminal locations are shaded in the figure. Each process is modelled by an EFSM, with updates representing the atomic statements in the algorithm. For example, update $c_0 = 0$ in EFSM source corresponds to program location s_1 in Algorithm 1: it checks whether c_0 is equal to zero and does not change any variable values. The update $x_0' = x_0'$ assigns a new number to x_0 from its domain, regardless of its previous value.

The model is parametrised by the number n of sieve processes, and the maximum number m generated by the source. The variable set of the system is $\mathrm{vars}(\mathcal{E}) = \{x_0, \ldots, x_n, c_1, \ldots, c_n\}$, and all variables have the same domain $0, \ldots, m$. The range of feasible values for m depends on the number of sieve processes. For example, for $n = 3$ there are three sieve processes for the first three primes 2, 3, and 5. Then the smallest number incorrectly classified as a prime is 49, so m should not be greater than 48.

4 Unfolding Semantics

This section gives a semantics of extended finite-state machines in terms of ordinary finite-state machines (FSM) interacting in lock-step synchronisation. Sect. 4.1 defines the FSM model used, and section 4.2 defines concepts needed to convert variables to states. Then section 4.3 describes the process of converting EFSMs to FSMs.

4.1 Finite-State Machines

Finite-state machines interact using *events*, which are taken from a finite alphabet Σ. In addition, the *silent event* $\tau \notin \Sigma$ is used. It is not included in the alphabet Σ unless explicitly mentioned using the notation $\Sigma_\tau = \Sigma \cup \{\tau\}$. Further, Σ^* is the set of all finite traces of events from Σ, including the *empty trace* ε. The concatenation of two traces $s, t \in \Sigma^*$ is written as st. A trace $s \in \Sigma^*$ is called a *prefix* of $t \in \Sigma^*$, written $s \sqsubseteq t$, if $t = su$ for some $u \in \Sigma^*$.

Definition 3 A *finite-state machine (FSM)* is a tuple $G = \langle \Sigma_G, Q, \rightarrow, Q^\circ, Q^\omega \rangle$, where $\Sigma_G \subseteq \Sigma$ is a finite set of events, called the *event alphabet* of G, Q is a finite set of *states*, $\rightarrow \subseteq Q \times (\Sigma_G \cup \{\tau\}) \times Q$

is the *state transition relation*, $Q^\circ \subseteq Q$ is the set of *initial states*, and $Q^\omega \subseteq Q$ is the set of *terminal states*.

The transition relation is written in infix notation $x \xrightarrow{\sigma} y$, and is extended to events not in the event alphabet by letting $x \xrightarrow{\sigma} x$ for all $\sigma \in \Sigma \setminus \Sigma_G$. It is further extended to traces in Σ_τ^* by $x \xrightarrow{\varepsilon} x$ for all $x \in Q$, and $x \xrightarrow{s\sigma} z$ if $x \xrightarrow{s} y$ and $y \xrightarrow{\sigma} z$ for some $y \in Q$. The transition relation is also defined for state sets $X, Y \subseteq Q$, for example $X \xrightarrow{s} y$ means $x \xrightarrow{s} y$ for some $x \in X$, and $G \xrightarrow{s} x$ stands for $Q^\circ \xrightarrow{s} x$.

Unlike EFSMs, the FSMs considered here interact using lock-step synchronisation [11]. The composition of FSMs can only execute an event if all synchronised FSMs are in a state enabling that event. An FSM always enables any event not in its alphabet.

Definition 4 Let $G_1 = \langle \Sigma_1, Q_1, \rightarrow_1, Q_1^\circ, Q_1^\omega \rangle$ and $G_2 = \langle \Sigma_2, Q_2, \rightarrow_2, Q_2^\circ, Q_2^\omega \rangle$ be two FSMs. The *synchronous composition* of G_1 and G_2 is

$$G_1 \parallel G_2 = \langle \Sigma_1 \cup \Sigma_2, Q_1 \times Q_2, \rightarrow, Q_1^\circ \times Q_2^\circ, Q_1^\omega \times Q_2^\omega \rangle \ , \tag{3}$$

where

- $(x_1, x_2) \xrightarrow{\sigma} (y_1, y_2)$ if $\sigma \neq \tau$ and $x_1 \xrightarrow{\sigma}_1 y_1$ and $x_2 \xrightarrow{\sigma}_2 y_2$;
- $(x_1, x_2) \xrightarrow{\tau} (y_1, x_2)$ if $x_1 \xrightarrow{\tau}_1 y_1$;
- $(x_1, x_2) \xrightarrow{\tau} (x_1, y_2)$ if $x_2 \xrightarrow{\tau}_2 y_2$.

4.2 Variables and Valuations

The state space of an EFSM system is not only determined by its locations, but also by its variables and their possible values.

For an update $p \in \Pi_V$, the term $\mathrm{vars}(p)$ denotes the set of all variables that occur in p, and $\mathrm{vars}'(p)$ denotes the set of all variables modified by p. For example, if $p \equiv x' = y + 1$ then $\mathrm{vars}(p) = \{x, y\}$, and $\mathrm{vars}'(p) = \{x\}$. When a transition $x \xrightarrow{p} y$ occurs, the variables in $\mathrm{vars}'(p)$ may change as specified by the update p, whereas all other variables remain unchanged. An update p with $\mathrm{vars}'(p) = \emptyset$ is called a *pure guard*. Its execution leaves all variables unchanged.

Given an EFSM $E = \langle V, Q, \rightarrow, Q^\circ, Q^\omega \rangle$, its set of variables is $\mathrm{vars}(E) = V$, and the variable set of an EFSM system $\mathcal{E}$ is $\mathrm{vars}(\mathcal{E}) = \bigcup_{E \in \mathcal{E}} \mathrm{vars}(E)$.

Given a set $V = \{v_1, \ldots, v_n\}$ of variables, its domain $\mathrm{dom}(V) = \mathrm{dom}(v_1) \times \cdots \times \mathrm{dom}(v_n)$ determines all possible combinations of variable values, and thus the set of possible system states. An element of $\mathrm{dom}(V)$ is denoted by $\bar{v} = (\bar{v}_0, \ldots, \bar{v}_n)$ with $\bar{v}_i \in \mathrm{dom}(v_i)$.

The elements $\bar{v} \in \mathrm{dom}(V)$ are also considered as *valuations*:

$$p(\bar{v}) \in \{\mathrm{true}, \mathrm{false}\} \tag{4}$$

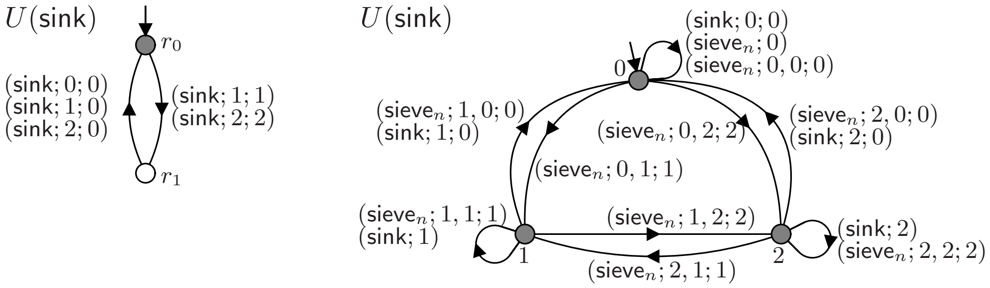


Figure 2: Unfolding of sink in Sieve of Eratosthenes example.

denotes the truth value of update $p \in \Pi_V$ when the variable values are given by $\bar{v}$. For $\bar{v} \in \mathrm{dom}(V)$, the value of the variable $v_i \in V$ in $\bar{v}$ is denoted by $\bar{v}[v_i]$. The set of variables assigned by a valuation $\bar{v}$ is denoted by $\mathrm{vars}(\bar{v})$. The *empty valuation* with $\mathrm{vars}(\bar{v}) = \emptyset$ is also denoted $\bar{v} = \emptyset$. For two sets of variables $W \subseteq V$, the valuation $\bar{v}\colon V \to D$ is said to be an *extension* of $\bar{w}\colon W \to D$, written $\bar{w} \le \bar{v}$, if $\bar{w}[w] = \bar{v}[w]$ for each $w \in W$.

4.3 Converting EFSMs to FSMs

The straightforward method [3] to convert an EFSM to an FSM creates a single FSM with states for each combination of a location and variable values. While this works well for symbolic state space exploration, the compositional verification method [9] pursued here demands a *compositional* model consisting of several FSMs. Therefore, the method proposed in the following preserves the compositional structure of an EFSM system by creating one FSM for each EFSM and for each variable.

An EFSM is converted to an FSM, which uses the EFSM locations as states and has the same transitions, except that they are labelled with events instead of updates. Each valuation that satisfies the update is represented by its own event.

Definition 5 Let $E = \langle V, Q, \to, Q^\circ, Q^\omega \rangle$ be an EFSM. The *unfolded FSM* of E is $U(E) = \langle \Sigma_E, Q, \to_U, Q^\circ, Q^\omega \rangle$ where,

- $\Sigma_E = \{\, (E;\hat{v};\hat{w}) \mid x \xrightarrow{p} y,\ \hat{v} \in \mathrm{dom}(\mathrm{vars}(p)),\ \hat{w} \in \mathrm{dom}(\mathrm{vars}'(p)) \,\}$;
- $x \xrightarrow{(E;\hat{v};\hat{w})}_U y$ if there exists a transition $x \xrightarrow{p} y$ in E such that $\hat{v} \in \mathrm{dom}\ (\mathrm{vars}(p))$, $\hat{w} \in \mathrm{dom}(\mathrm{vars}'(p))$, and $p(\hat{v},\hat{w}) = \mathrm{true}$.

An EFSM update p is replaced by FSM events $(E;\hat{v};\hat{w})$ for all valuations $\hat{v}$ defined over the variables of p and $\hat{w}$ defined over the next-state variables of p, such that $\hat{v}$ and $\hat{w}$ together satisfy p. Note that $\mathrm{vars}(\hat{w}) \subseteq \mathrm{vars}(\hat{v})$ due to the definition of $\mathrm{vars}(p)$ and $\mathrm{vars}'(p)$. Pure guards produce events $(E;\hat{v};\emptyset)$, which are simply written as $(E;\hat{v})$ in the following.

Example 1 Consider EFSM sink in the Sieve of Eratosthenes example shown in figure 1, assuming $m = 2$, i.e., $\mathrm{dom}(c_n) = \{0, 1, 2\}$, and $c_n^\circ = 0$. The update $c_n > 0$ in sink results in the unfolded events $\Sigma^1_{\mathsf{sink}} = \{(\mathsf{sink}; 1), (\mathsf{sink}; 2)\}$, and update $c'_n = 0$ results in events $\Sigma^1_{\mathsf{sink}} = \{(\mathsf{sink}; 0; 0), (\mathsf{sink}; 1; 0), (\mathsf{sink}; 2; 0)\}$. Thus, the unfolded event set of sink is $\Sigma_{\mathsf{sink}} = \Sigma^1_{\mathsf{sink}} \cup \Sigma^2_{\mathsf{sink}}$, and the unfolded FSM $U(\mathsf{sink})$ is shown in figure 2 to the left.

The state space of an EFSM system is not only determined by its locations, but also by its variables. Therefore a second set of FSMs, called *variable FSMs*, is used to keep track of the variable values and ensure the correct sequencing of the transitions in the unfolded FSM system.

Definition 6 Let $\mathcal{E} = \{E_1, \ldots, E_n\}$ be an EFSM system. The *variable FSM* of $v \in \mathrm{vars}(\mathcal{E})$ is $U_{\mathcal{E}}(v) = \langle \Sigma_v, \mathrm{dom}(v), \rightarrow_v, \{\bar{v}^\circ\}, \mathrm{dom}(v) \rangle$ where,

- $\Sigma_v = \{\, (E_i; \hat{v}; \hat{w}) \in \Sigma_{E_i} \mid v \in \mathrm{vars}(\hat{v}) \,\}$;
- $\hat{v}[v] \xrightarrow{(E_i;\hat{v};\hat{w})}_v \hat{v}[v]$ if $v \in \mathrm{vars}(\hat{v}) \setminus \mathrm{vars}(\hat{w})$;
- $\hat{v}[v] \xrightarrow{(E_i;\hat{v};\hat{w})}_v \hat{w}[v]$ if $v \in \mathrm{vars}(\hat{w})$.

Example 2 Consider the variable c_n in the Sieve of Eratosthenes example. It occurs in sink and sieve_n, so these EFSMs determine the event alphabet of $U_{\mathcal{E}}(c_n)$. First, all transitions in sink mention c_n, so the full alphabet Σ_{sink} from example 1 is included. Next, sieve_n contains two updates associated with c_n. The update $c_n = 0$ produces one unfolded event $\Sigma^1_{c_n} = \{(\mathsf{sieve}_n; 0)\}$. Further, the update $c'_n = x_n$ with $\mathrm{vars}(c'_n = x_n) = \{c_n, x_n\}$ and $\mathrm{vars}'(c'_n = x_n) = \{c_n\}$ produces events of the form $(\mathsf{sieve}_n; c_n, x_n; c_n)$. Again assuming $\mathrm{dom}(c_n) = \mathrm{dom}(x_n) = \{0, 1, 2\}$, these are:

$$\begin{aligned}\Sigma^2_{c_n} = \{\, & (\mathsf{sieve}_n; 0, 0; 0), (\mathsf{sieve}_n; 0, 1; 1), (\mathsf{sieve}_n; 0, 2; 2),\\ & (\mathsf{sieve}_n; 1, 0; 0), (\mathsf{sieve}_n; 1, 1; 1), (\mathsf{sieve}_n; 1, 2; 2),\\ & (\mathsf{sieve}_n; 2, 0; 0), (\mathsf{sieve}_n; 2, 1; 1), (\mathsf{sieve}_n; 2, 2; 2) \,\} \, .\end{aligned} \tag{5}$$

This gives $\Sigma_{c_n} = \Sigma_{\mathsf{sink}} \cup \Sigma^1_{c_n} \cup \Sigma^2_{c_n}$ and the variable FSM $U_{\mathcal{E}}(c_n)$ as shown in figure 2 to the right.

The variable FSMs are defined in the context of an EFSM system, as they depend on all EFSMs using the variable. The overall behaviour of an EFSM system is obtained by applying the unfolding method to all its EFSMs and variables.

Definition 7 Let $\mathcal{E} = \{E_1, \ldots, E_n\}$ be an EFSM system. The *unfolding* of $\mathcal{E}$ is the FSM

$$U(\mathcal{E}) = \mathop{\Big\|}_{i=1}^{n} U(E_i) \;\, \| \mathop{\Big\|}_{v \in \mathrm{vars}(\mathcal{E})} U_{\mathcal{E}}(v) \, . \tag{6}$$

5 Compositional Nonblocking Verification

This working paper concerns verification of the *nonblocking* property used in supervisory control theory of discrete event systems [20], which can capture the absence of livelocks and deadlocks. A system is nonblocking if it is possible to reach a terminal state from every reachable state. For finite-state systems, nonblocking is equivalent to termination under an implicit *strong fairness* assumption stating that “whenever a transition can occur infinitely often, it occurs infinitely often” [2].

Definition 8 [20] An FSM $G = \langle \Sigma, Q, \to, Q^\circ, Q^\omega \rangle$ is *nonblocking* if, for every $s \in \Sigma_\tau^*$ and every $x \in Q$ such that $G \xrightarrow{s} x$, there exists $t \in \Sigma_\tau^*$ such that $x \xrightarrow{t} Q^\omega$.

Definition 9 An EFSM system $\mathcal{E}$ is nonblocking if the unfolding $U(\mathcal{E})$ is nonblocking. An EFSM E is nonblocking if the EFSM system $\{E\}$ is nonblocking.

The straightforward approach to check whether a system

$$P_1 \parallel P_2 \parallel \cdots \parallel P_n \tag{7}$$

is nonblocking is to explicitly construct the synchronous composition and check for each reachable state whether it is possible to reach a terminal state. This can be done using CTL model checking, and models of substantial size can be analysed if the state space is represented symbolically [17]. Yet, the technique remains limited by the amount of memory available to store representations of the synchronous composition.

In an attempt to alleviate this state-space explosion problem, *compositional* verification [9] seeks to rewrite individual system components and, for example, replace P_1 in (7) by a simpler *abstraction* P_1', to analyse the simpler system

$$P_1' \parallel P_2 \parallel \cdots \parallel P_n \tag{8}$$

Several abstraction methods that preserve the nonblocking property are known [9, 15, 22]. Based on these methods, compositional verification algorithms [9, 22] repeatedly simplify system components, compose subsystems and simplify them again, until the system is simple enough to be verified directly. These methods have been developed and used successfully to verify several large FSM models [9].

To assess the applicability of compositional verification for EFSM models with data dependency, the Distributed Sieve of Eratosthenes has been modelled and verified using the Discrete Event Systems tool *Supremica* [1]. Supremica converts the EFSM model to a collection of unfolded FSMs [14], which are then verified using an implementation of the compositional nonblocking algorithm [9].

Table 1 shows the results of these experiments for different prime number sieves, where n is the number of sieve processes, and m is the largest number generated by the source. The table shows in each case the number of events and transitions in the unfolded FSM model, and the number of reachable states in its synchronous composition; it furthermore shows the number of states of the

Table 1: Experimental Results for Distributed Sieve of Eratosthenes.

n	m	**Events**	**Transitions**	**State space**	**Peak**	**Time**	**Memory**
2	24	144	10,460	$7.59{\cdot}10^{7}$	27	0.22 s	90.4 MB
3	48	369	53,345	$3.45{\cdot}10^{13}$	51	0.81 s	147.7 MB
4	120	1,122	404,504	$1.18{\cdot}10^{20}$	123	3.04 s	260.3 MB
5	168	1,899	958,193		171	7.47 s	331.0 MB
6	288	3,804	3,288,972		291	37.67 s	532.4 MB

largest FSM encountered during compositional verification (Peak), and the approximate runtime and memory usage of compositional verification. The experiments were run on a standard laptop computer using a single core 2.4 GHz CPU.

Supremica successfully verifies the Distributed Sieve of Eratosthenes to be nonblocking for of to $n = 6$ sieve processes. It has also been attempted using Supremica to verify the model symbolically with BDDs [17], but this was unsuccessful for $n \geq 5$ sieve processes, so the number of reachable states is not known for the larger models.

This experiment suggests that compositional verification is a promising approach to verify large EFSM systems, with the peak number of states only growing proportionally to the parameter m. However, the number of events in the unfolded FSM model grows with nm, and the number of transitions grows with nm^2. At $n = 6$, the construction of the unfolded FSM model already takes substantially longer than its verification. To avoid the construction of a growing FSM model, the following section proposes an alternative approach to perform compositional verification directly on the EFSMs.

6 Abstraction Methods

Compositional verification repeats two basic operations while verifying a system: either individual components are simplified or, if this is not possible, two or more components are composed. Sect. 6.1 and 6.2 below describe the method of composition and the related method of unfolding local variables, then section 6.3 introduces the principle of simplification, and section 6.4 and 6.5 propose two methods to simplify EFSMs.

6.1 Partial Composition

Composition is the simplest step in compositional verification. It is always possible to replace some components of an EFSM system by their composition. This operation does not reduce the state space, but it is necessary when all other means of simplification have been exhausted. The following result, albeit technical, follows directly from the definitions. The unfolded FSMs before and after partial composition are not only equivalent with respect to nonblocking, but identical up to renaming of events. The proof can be found in Appendix A.

Proposition 1 Let $\mathcal{E} = \{E_1, \dots, E_n\}$ be an EFSM system, and $\mathcal{F} = \{E_1 \parallel E_2, E_3, \dots, E_n\}$. Then $U(\mathcal{E})$ is nonblocking if and only if $U(\mathcal{F})$ is nonblocking.

6.2 Partial Unfolding

Similar to partial composition, partial unfolding is the process of removing a variable from an EFSM and expanding its values into locations.

Definition 10 Let $E = \langle V, Q, \rightarrow, Q^\circ, Q^\omega \rangle$ be an EFSM, and let $z \in V$. The result of *partially unfolding* z in E is the EFSM $E \setminus z = \langle V, Q \times \mathrm{dom}(z), \rightarrow_{-z}, Q^\circ \times \{\bar{z}^\circ\}, Q^\omega \times \mathrm{dom}(z) \rangle$ where

$$(x, a) \xrightarrow{\exists z \exists z' (p \wedge z = a \wedge z' = b)}_{-z} (y, b) \tag{9}$$

for all $a, b \in \mathrm{dom}(z)$ such that $x \xrightarrow{p} y$, and such that $z \notin \mathrm{vars}'(p)$ implies $a = b$.

A variable is called *local* in an EFSM system, if it appears in only one component. Local variables can be removed by partial unfolding, as they are not needed for interaction with any other component. The following result confirms that partial unfolding of a local variable preserves the nonblocking property of an EFSM system. The proof is similar to that of proposition 1 and shows that the unfolded FSMs of E_1 and $E_1 \setminus z$ are identical up to renaming of events. It can be found in Appendix B.

Proposition 2 Let $\mathcal{E} = \{E_1, \dots, E_n\}$ be an EFSM system, $z \in \mathrm{vars}(E_1) \setminus \bigcup_{i=2}^{n} \mathrm{vars}(E_i)$, and $\mathcal{F} = \{E_1 \setminus z, E_2, \dots, E_n\}$. Then $U(\mathcal{E})$ is nonblocking if and only if $U(\mathcal{F})$ is nonblocking.

Partial unfolding removes local variables at the price of an increase in the number of locations. Its application may be deferred in favour of other methods. On the other hand, partial unfolding often simplifies or removes some updates, making it possible to apply the abstraction methods following below, which reduce the state space.

6.3 Conflict Equivalence

Compositional reasoning is based on the idea of replacing a component P_k in a larger system (7) by an equivalent component P'_k. The best known equivalence to support compositional nonblocking verification of FSMs is *conflict equivalence* [16]. In the following, this concept is extended to EFSMs.

The idea of conflict equivalence is derived from process-algebraic testing theory [8], which defines equivalences relating processes based on the results of *tests*. Two processes are considered as equivalent if the responses of all tests are equal. Here, a test's result is the observation whether or not it is nonblocking in composition with the process under test. The following definition is generalised for arbitrary *components*, which can be either FSMs or EFSMs.

Definition 11 [16] Two components P_1 and P_2 are *conflict equivalent*, written $P_1 \simeq_{\mathrm{conf}} P_2$, if for any component T, it holds that $P_1 \parallel T$ is nonblocking if and only if $P_2 \parallel T$ is nonblocking.

Conflict equivalence guarantees that, if a component P_k is replaced by a conflict equivalent abstraction P'_k, the abstraction will produce the same verification result, in combination with every possible "remainder of the system", T, as would the original component P_k. The following result confirms that conflict equivalent components of an EFSM system can be replaced without affecting the nonblocking property. This is the key property of conflict equivalence, which follows from its congruence properties [16]. The proof is given in Appendix C.

Proposition 3 Let $\mathcal{E} = \{E_1, \ldots, E_n\}$ and $\mathcal{F} = \{F_1, E_2, \ldots, E_n\}$ be EFSM systems such that $E_1 \simeq_{\text{conf}} F_1$. Then $U(\mathcal{E})$ is nonblocking if and only if $U(\mathcal{F})$ is nonblocking.

6.4 Symbolic Observation Equivalence

Bisimulation and *observation equivalence* [18] are standard examples of branching equivalences. They are known to preserve all temporal logic properties [4], including nonblocking. Observation equivalence alone is responsible for a substantial state-space reduction in compositional nonblocking verification of FSMs [9]. Both bisimulation and observation equivalence have been generalised for value-passing processes [10, 13]. In this section, observation equivalence is extended to be applicable for EFSMs, and *symbolic observation equivalence* is proposed.

The most basic branching equivalence is *bisimulation*, which keeps track of the complete branching of process behaviour.

Definition 12 Let $E = \langle V, Q_E, \rightarrow_E, Q^\circ_E, Q^\omega_E \rangle$ and $F = \langle V, Q_F, \rightarrow_F, Q^\circ_F, Q^\omega_F \rangle$ be two EFSMs. A relation $\approx\, \subseteq Q_E \times Q_F$ is called a *symbolic bisimulation* between E and F if the following holds for all $x_E \in Q_E$ and $x_F \in Q_F$ such that $x_E \approx x_F$:

- if $x_E \overset{p_E}{\rightarrow}_E y_E$, then there exists $y_F \in Q_F$ such that $x_F \overset{p_F}{\rightarrow}_F y_F$ and p_E logically implies p_F and $y_E \approx y_F$;
- if $x_F \overset{p_F}{\rightarrow}_F y_F$, then there exists $y_E \in Q_E$ such that $x_E \overset{p_E}{\rightarrow}_E y_E$ and p_F logically implies p_E and $y_E \approx y_F$;
- $x_E \in Q^\omega_E$ if and only if $x_F \in Q^\omega_2$.

E and F are *symbolically bisimilar*, written $E \approx F$, if there exists a symbolic bisimulation $\approx$ between E and F such that, for each $x^\circ_E \in Q^\circ_F$ there exists $x^\circ_F \in Q^\circ_F$ such that $x^\circ_E \approx x^\circ_F$, and vice versa.

While symbolic bisimulation as defined implies conflict equivalence, the definition is restrictive as it requires syntactically equivalent updates for locations to be equivalent. For FSMs, observation equivalence is the natural extension of bisimulation. In observation equivalence, the transition relation $\rightarrow$ is replaced by its extension $\Rightarrow$ to allow for silent transitions before or after an event occurrence. To extend this idea for EFSMs, the first step is to define the extended transition relation $\Rightarrow$ for EFSMs.

Definition 13 Let $E = \langle V, Q, \rightarrow, Q^\circ, Q^\omega \rangle$ be an EFSM.

- For $x, y \in Q$ and $\bar{v} \in \mathrm{dom}(\mathrm{vars}(E))$, the relation $x \overset{\bar{v}}{\Rightarrow} y$ denotes the existence of a path
$$x = x_0 \xrightarrow{p_1} x_1 \xrightarrow{p_2} \cdots \xrightarrow{p_n} x_n = y \ , \tag{10}$$
such that $\mathrm{vars}'(p_i) = \emptyset$ and $p_i(\bar{v}) = \mathrm{true}$ for each $1 \leq i \leq n$.

- For $x, y \in Q$ and $\bar{v}, \bar{w} \in \mathrm{dom}(\mathrm{vars}(E))$, the relation $x \overset{\bar{v},\bar{w}}{\Longrightarrow} y$ means that there exist states $x_1, y_1 \in Q$ such that
$$x \overset{\bar{v}}{\Rightarrow} x_1 \xrightarrow{p} y_1 \overset{\bar{w}}{\Rightarrow} y \ , \tag{11}$$
where $p(\bar{v}, \bar{w}) = \mathrm{true}$ and $\bar{w}_{|\mathrm{vars}(E)\setminus\mathrm{vars}'(p)} \leq \bar{v}$.

- For $x \in Q$, the relation $E \Rightarrow x$ denotes the existence of $x^\circ \in Q^\circ$ such that $x^\circ \overset{\bar{v}^\circ}{\Rightarrow} x$.

The notation $x \overset{\bar{v}}{\Rightarrow} y$ means that it is possible for an EFSM to move from location x to y while the variables remain constant at $\bar{v}$, and $x \overset{\bar{v},\bar{w}}{\Longrightarrow} y$ means that it is possible to move from x to y with a single change of variable values from $\bar{v}$ to $\bar{w}$. The condition $\bar{w}_{|\mathrm{vars}(E)\setminus\mathrm{vars}'(p)} \leq \bar{v}$ ensures that variables not affected by the update p remain unchanged. With this symbolic definition of the extended transition relation, symbolic observation equivalence is defined as follows.

Definition 14 Let $E = \langle V, Q_E, \rightarrow_E, Q_E^\circ, Q_E^\omega \rangle$ and $F = \langle V, Q_F, \rightarrow_F, Q_F^\circ, Q_F^\omega \rangle$ be two EFSMs. A relation $\sim \subseteq Q_E \times Q_F$ is called a *symbolic observation equivalence* between E and F if the following holds for all $x_E \in Q_E$ and $x_F \in Q_F$ such that $x_E \sim x_F$:

- if $x_E \overset{\bar{v},\bar{w}}{\Longrightarrow}_E y_E$, then there exists $y_F \in Q_F$ such that $x_F \overset{\bar{v},\bar{w}}{\Longrightarrow}_F y_F$ and $y_E \sim y_F$;
- if $x_F \overset{\bar{v},\bar{w}}{\Longrightarrow}_F y_F$, then there exists $y_E \in Q_E$ such that $x_E \overset{\bar{v},\bar{w}}{\Longrightarrow}_E y_E$ and $y_E \sim y_F$;
- $x_E \overset{\bar{v}}{\Rightarrow}_E Q_E^\omega$ if and only if $x_F \overset{\bar{v}}{\Rightarrow}_F Q_F^\omega$.

E and F are *symbolically observation equivalent*, written $E \sim F$, if there exists a symbolic observation equivalence $\sim$ between E and F such that, for each $x_E^\circ \in Q_E$ such that $E \overset{\bar{v}^\circ}{\Rightarrow} x_E^\circ$ there exists $x_F^\circ \in Q_F$ such that $F \overset{\bar{v}^\circ}{\Rightarrow} x_F^\circ$ and $x_E^\circ \sim x_F^\circ$, and vice versa.

Two locations are symbolically observation equivalent, if they can reach equivalent successors by means of the extended transition relation $\Rightarrow$. Symbolic observation equivalence is closely related to observation equivalence of the unfolded FSMs, which is known to imply conflict equivalence [9]. The following result, with proof in Appendix D, confirms that symbolically observation equivalent EFSMs are conflict equivalent. In combination with proposition 3, it is clear that components in an EFSM system can be replaced by symbolically observation equivalent abstractions without affecting the nonblocking property of the system.

Proposition 4 Let E_1 and F_1 be two EFSMs. If $E_1 \sim F_1$ then $E_1 \simeq_{\mathrm{conf}} F_1$.

6.5 Active Events Rule

While observation equivalence reduces the size of FSMs significantly and is easy to implement, it is not the best possible equivalence for nonblocking verification [16]. Several abstraction rules preserving conflict equivalence of FSMs are known [9, 15] that extend beyond observation equivalence. This section extends one of these rules, namely the *Active Events Rule* [9], to EFSMs.

The Active Events Rule for FSMs allows to merge states with the same sets of enabled events, provided they are also *incoming equivalent*.

Definition 15 Let $E = \langle V, Q, \to, Q^\circ, Q^\omega \rangle$ be an EFSM. The *incoming equivalence* relation $\sim_{\text{inc}} \subseteq Q \times Q$ of E is defined such that $y_1 \sim_{\text{inc}} y_2$ if

- $E \Rightarrow y_1$ if and only if $E \Rightarrow y_2$;
- for all $x \in Q$ and all $\bar{v}, \bar{w} \in \text{dom}(\text{vars}(E))$, it holds that $x \overset{\bar{v},\bar{w}}{\Longrightarrow} y_1$ with $\bar{v} \neq \bar{w}$ or $x \neq y_1$ implies $x \overset{\bar{v},\bar{w}}{\Longrightarrow} y_2$, and vice versa.

Two incoming equivalent locations have exactly the same incoming transitions with equivalent updates and equal source locations. Unlike with FSMs, selfloops $x \overset{\bar{v}}{\Rightarrow} x$ are excluded, because by definition 13, $x \overset{\bar{v}}{\Rightarrow} x$ holds for every location x, and including them would require all incoming equivalent locations to be linked to each other.

Definition 16 Let $E = \langle V, Q, \to, Q^\circ, Q^\omega \rangle$ be an EFSM. The *active events equivalence* relation $\sim_{\text{act}} \subseteq Q \times Q$ of E is defined such that $x_1 \sim_{\text{act}} x_2$ if

- for all $\bar{v}, \bar{w} \in \text{dom}(\text{vars}(E))$, it holds that $x_1 \overset{\bar{v},\bar{w}}{\Longrightarrow} y_1$ for some $y_1 \in Q$ such that $\bar{v} \neq \bar{w}$ or $x_1 \neq y_1$, if and only if $x_2 \overset{\bar{v},\bar{w}}{\Longrightarrow} y_2$ for some $y_2 \in Q$ such that $\bar{v} \neq \bar{w}$ or $x_2 \neq y_2$;
- for all $\bar{v} \in \text{dom}(\text{vars}(E))$ it holds that $x_1 \overset{\bar{v}}{\Rightarrow} Q^\omega$ if and only if $x_2 \overset{\bar{v}}{\Rightarrow} Q^\omega$.

Two locations are active events equivalent if they have exactly the same outgoing transitions, independently of their target locations. Selfloops are only considered if they are considered in incoming equivalence. Based on these concepts, the *Active Events Rule* is defined in the same way as for FSMs and says that, two locations that are both incoming and active events equivalent are conflict equivalent and can be merged.

The idea is that, for conflict equivalence only the traces leading to terminal states are relevant. If two states are reached in exactly the same way and have exactly the same transitions enabled, then the nondeterministic choice between these two states can be deferred by one step and the states can be merged. Technically, this is done by the standard construction of a quotient automaton [3]. Prop. 5 describes the Active Events Rule formally, and the proof can be found in Appendix E.

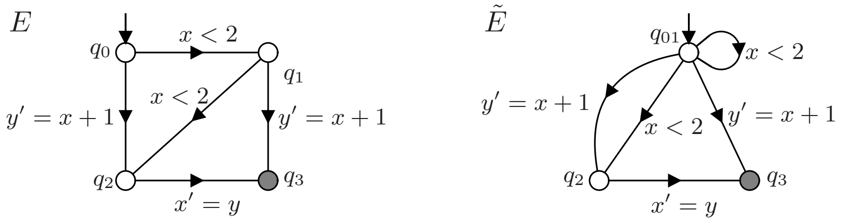


Figure 3: Example of Active Events Rule.

Definition 17 Let $E = \langle V, Q, \rightarrow, Q^\circ, Q^\omega \rangle$ be an EFSM, and let $\sim \subseteq Q \times Q$ be an equivalence relation. The *quotient EFSM* of E modulo $\sim$ is $E/\!\sim = \langle V, Q/\!\sim, \rightarrow/\!\sim, \tilde{Q}^\circ, \tilde{Q}^\omega \rangle$, where

$$\rightarrow/\!\sim = \{ ([x], p, [y]) \mid x \xrightarrow{p} y \} ; \tag{12}$$

$$\tilde{Q}^\circ = \{ [x] \mid x \in Q^\circ \} ; \tag{13}$$

$$\tilde{Q}^\omega = \{ [x] \mid x \in Q^\omega \} . \tag{14}$$

Here, $[x] = \{ x \in Q \mid x' \sim x \}$ denotes the *equivalence class* of $x \in Q$ with respect to $\sim$, and $Q/\!\sim = \{ [x] \mid x \in Q \}$ is the set of equivalence classes modulo $\sim$.

Proposition 5 Let $E_1 = \langle V, Q, \rightarrow, Q^\circ, Q^\omega \rangle$ be an EFSM, and let $\sim \subseteq Q \times Q$ be an equivalence relation such that $\sim \subseteq \sim_{\text{inc}} \cap \sim_{\text{act}}$, where $\sim_{\text{inc}}$ and $\sim_{\text{act}}$ are the incoming and active events equivalences of E_1. Then $E_1 \simeq_{\text{conf}} E_1/\!\sim$.

Example 3 Consider EFSM E in figure 3, and assume $x^\circ = y^\circ = 0$. Given that $q_0 \overset{x<2}{\Longrightarrow} q_0$ by definition 13, locations q_0 and q_1 are both reached from the initial location q_0 when $x < 2$, and this establishes $q_0 \sim_{\text{inc}} q_1$. Furthermore, both locations q_0 and q_1 have outgoing non-selfloop transitions with updates $y' = x + 1$ and $x < 2$, which shows that $q_0 \sim_{\text{act}} q_1$. By the Active Events Rule, these locations are conflict equivalent and can be merged, resulting in $\tilde{E}$ in figure 3. Yet, q_0 and q_1 are not observation equivalent as the transitions $y' = x + 1$ from q_0 and q_1 lead to different locations that are not equivalent.

7 Example Revisited

In this section, the compositional verification procedure is applied to the Sieve of Eratosthenes introduced in section 3. For illustration, the number of sieve processes is set to $n = 2$, and while the resultant sieve can recognise prime numbers up to 24, the range is restricted to $m = 7$. The system consists of four EFSMs source, sieve_1, sieve_2, and sink, shown in figure 1, and its unfolded state space has 2,385,179 reachable states.

None of the EFSMs in figure 1 can be simplified using either observation equivalence or active events, but some variables are local and can be partially unfolded. Unfolding x_0 in source results in

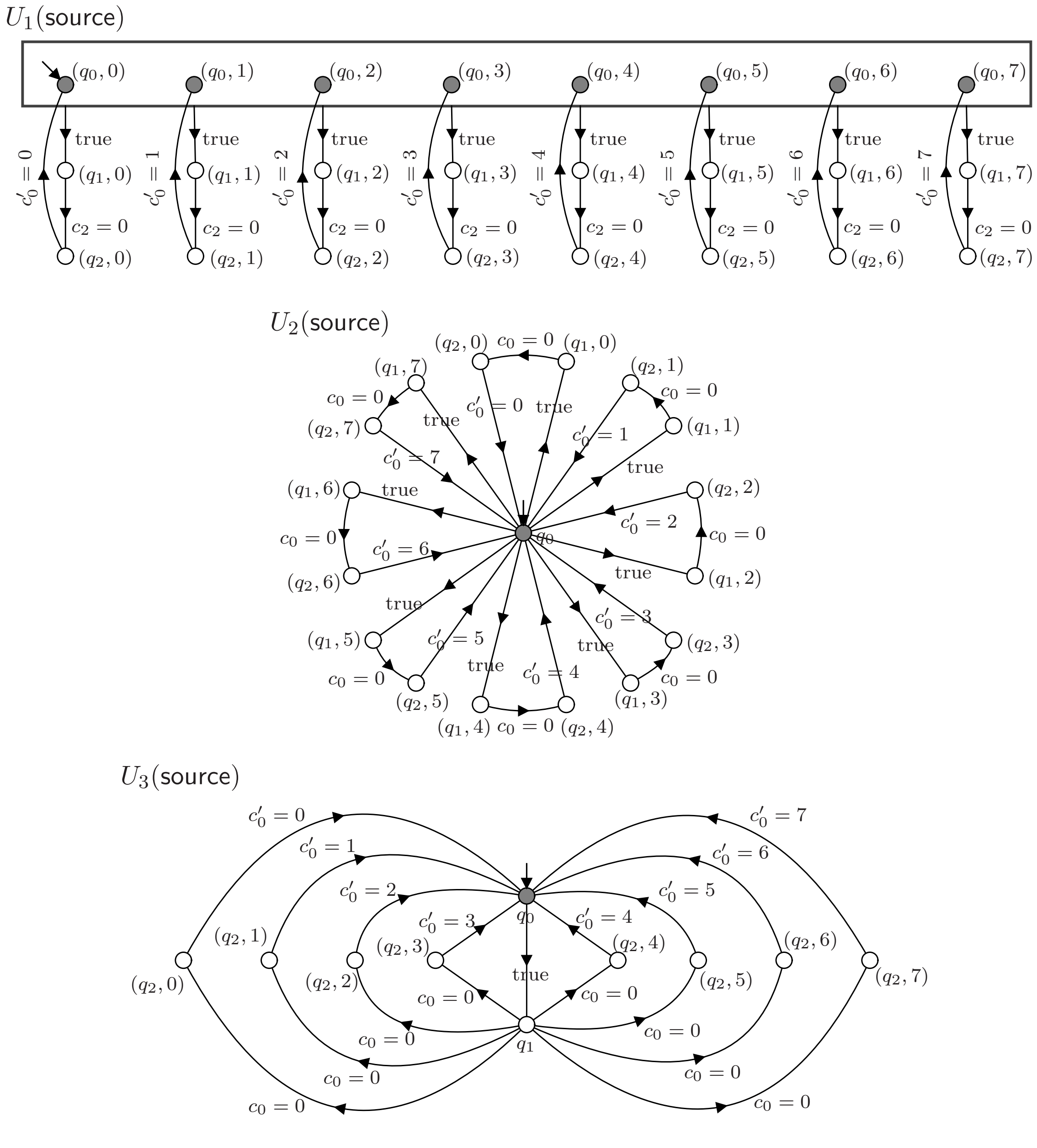


Figure 4: Abstractions of source in Sieve of Eratosthenes example.

the 24-location EFSM $U_1(\mathsf{source})$ shown in figure 4. For graphical clarity, the figure uses a *group node* to combine the locations (q_0, i) for $0 \leq i \leq 7$: each transition out of the box stands for eight different transitions with the same update and target location, one transition from each location in the group. Clearly, the locations (q_0, i) in the group all have exactly the same outgoing transitions, so they are bisimilar and can be merged into a single location q_0. This results in the abstraction $U_2(\mathsf{source})$, also shown in figure 4. Locations (q_1, i) for $0 \leq i \leq 7$ in $U_2(\mathsf{source})$ are incoming equivalent, as they all have the same incoming transition from location q_0 with update true, and active events equivalent, as they all have only one outgoing transition with update $c_0 = 0$. These locations can be merged using the Active Events Rule, resulting in the 10-location EFSM $U_3(\mathsf{source})$ in figure 4.

Next, the variable x_1 is local in sieve_1, and its unfolding results in a 49-location EFSM $U_1(\mathsf{sieve}_1)$, shown in figure 5. Observation equivalence simplifies this to an 18-location EFSM $U_2(\mathsf{sieve}_1)$, also shown in figure 5. Similarly, partial unfolding of x_2 in sieve_2 and observation equivalence result in a 21-location EFSM $U_2(\mathsf{sieve}_2)$. The sink EFSM cannot be simplified.

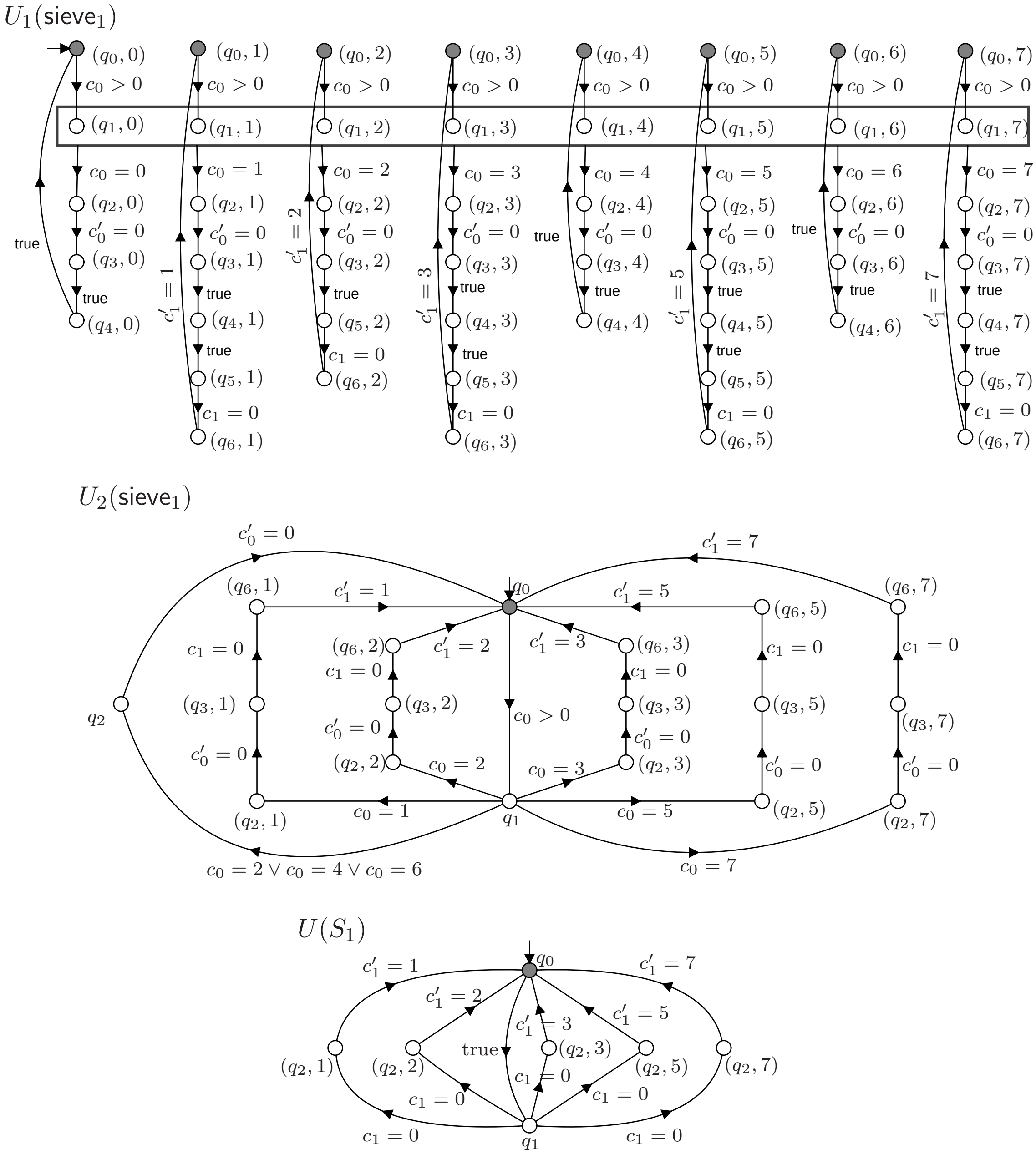
At this point, the system model consists of four EFSMs $U_3(\mathsf{source})$, $U_2(\mathsf{sieve}_1)$, $U_2(\mathsf{sieve}_2)$, and sink, and three variables c_0, c_1, and c_2. The number of reachable states in the unfolding is now 100,712. For compositional verification to proceed, some components need to be composed. After composing $U_2(\mathsf{sieve}_1)$ and $U_3(\mathsf{source})$, variable c_0 becomes local and can be unfolded. The resultant EFSM has 292 locations, and can be abstracted to 126 locations using observation equivalence, and further to 7 locations using the Active Events Rule. The resultant EFSM $U(S_1)$ is shown in figure 5. It is very similar to $U_3(\mathsf{source})$ in figure 4. The difference is that only the numbers 1, 2, 3, 5, and 7 are sent to the next stage of the pipeline, as 0, 4, and 6 are filtered out by the first sieve process.

Next, $U(S_1)$ and $U_2(\mathsf{sieve}_2)$ are composed, resulting in c_1 becoming a local variable. By unfolding c_1, a 207-location EFSM is obtained, which again is simplified to a 7 location EFSM $U(S_2)$ using observation equivalence and the Active Events Rule. $U(S_2)$ is the same as $U(S_1)$ except that c_1 is replaced by c_2. The abstraction of the initial segment of the pipeline does not change, as the first non-prime filtered out by sieve_2 is 9, but the source only produces numbers up to $m = 7$.

Now the system consists only of the EFSMs $U(S_2)$ and sink, and the variable c_2. Composition and unfolding results in a 27-state FSM, which is verified to be nonblocking. This is enough to conclude that the original system is nonblocking. Thus, a 2,385,179-state system has been verified to be nonblocking, and the largest component constructed in the process had 292 locations. The constructed abstractions only increase with the maximum number m produced by the source, not with the number n of sieve processes, showing that the method scales well as the parameters increase.

8 Conclusions

A framework for compositional nonblocking verification of reactive systems modelled as extended finite state machines (EFSM) is presented. The method is based on a generalisation of results about conflict equivalence for finite-state machines. State-space explosion is mitigated by gradually composing the components of a large system, and simplifying the intermediate results using the abstraction


$U_1(\mathsf{sieve}_1)$

$U_2(\mathsf{sieve}_1)$

$U(S_1)$

Figure 5: Abstractions of sieve_1 in Sieve of Eratosthenes example.

methods of symbolic observation equivalence and the Active Events Rule. The approach is demonstrated to scale well for an example of concurrent software.

Future work includes generalising other conflict-preserving abstraction rules, known to work well for FSMs, and adding them to the framework [9, 15]. Further, the method can likely be improved by combining it with known methods for variable abstraction and symbolic reasoning [3]. It is also possible to support event-based EFSM synchronisation, as it is already used in the underlying theory of conflict equivalence [16]. In addition, extension of the method for supervisor synthesis [20] for EFSMs is interesting.

References

[1] Knut Åkesson, Martin Fabian, Hugo Flordal, and Robi Malik. Supremica—an integrated environment for verification, synthesis and simulation of discrete event systems. In *Proceedings of the 8th International Workshop on Discrete Event Systems, WODES'06*, pages 384–385, Ann Arbor, MI, USA, July 2006.

[2] A. Arnold. *Finite Transitions Systems: Semantics of Communicating Systems*. Prentice-Hall, 1994.

[3] Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking*. MIT Press, 2008.

[4] Stephen D. Brookes and William C. Rounds. Behavioural equivalence relations induced by programming logics. In *Proceedings of 16th International Colloquium on Automata, Languages, and Programming, ICALP '83*, volume 154 of *LNCS*, pages 97–108. Springer-Verlag, 1983.

[5] Y. Chen and F. Lin. Modeling of discrete event systems using finite state machines with parameters. In *Proceedings of 2010 IEEE International Conference on Control Applications (CCA)*, pages 941–946, Anchorage, Alaska, USA, 2000.

[6] Kwang Ting Cheng and A. S. Krishnakumar. Automatic functional test generation using the extended finite state machine model. In *Proceedings of 30th ACM/IEEE Design Automation Conference*, pages 86–91, Dallas, TX, USA, 1993.

[7] Dennis Dams, Orna Grumberg, and Rob Gerth. Abstract interpretation of reactive systems: Abstractions preserving $\forall$CTL*, $\exists$CTL* and CTL*. In E.-R. Olderog, editor, *Proceedings of IFIP WG2.1/WG2.2/WG2.3 Working Conference on Programming Concepts, Methods and Calculi (PROCOMET)*, IFIP Transactions. Elsevier Science Publisher (North-Holland), Amsterdam, The Netherlands, June 1994.

[8] R. De Nicola and M. C. B. Hennessy. Testing equivalences for processes. *Theoretical Computer Science*, 34(1–2):83–133, November 1984.

[9] Hugo Flordal and Robi Malik. Compositional verification in supervisory control. *SIAM Journal of Control and Optimization*, 48(3):1914–1938, 2009.

[10] Matthew Hennessy and Huimin Lin. Symbolic bisimulations. *Theoretical Computer Science*, 138(2):353–389, 1995.

[11] C. A. R. Hoare. *Communicating Sequential Processes*. Prentice-Hall, 1985.

[12] John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, 2001.

[13] Hee-Hwan Kwak, Jin-Young Choi, Insup Lee, and Anna Philippou. Symbolic weak bisimulation for value-passing calculi. Technical Report MS-CIS-98-22, University of Pennsylvania, Department of Computer and Information Science, 1998.

[14] Robi Malik, Martin Fabian, and Knut Åkesson. Modelling large-scale discrete-event systems using modules, aliases, and extended finite-state automata. In *Proceedings of 18th IFAC World Congress*, pages 7000–7005, Milan, Italy, 2011.

[15] Robi Malik and Ryan Leduc. A compositional approach for verifying generalised nonblocking. In *Proceedings of 7th International Conference on Control and Automation, ICCA '09*, pages 448–453, Christchurch, New Zealand, December 2009.

[16] Robi Malik, David Streader, and Steve Reeves. Conflicts and fair testing. *International Journal of Foundations of Computer Science*, 17(4):797–813, 2006.

[17] K. L. McMillan. *Symbolic Model Checking*. Kluwer Academic Publishers, 1993.

[18] Robin Milner. *Communication and concurrency*. Series in Computer Science. Prentice-Hall, 1989.

[19] Sahar Mohajerani, Robi Malik, Simon Ware, and Martin Fabian. On the use of observation equivalence in synthesis abstraction. In *Proceedings of the 3rd IFAC Workshop on Dependable Control of Discrete Systems, DCDS 2011*, pages 84–89, Saarbrücken, Germany, 2011.

[20] Peter J. G. Ramadge and W. Murray Wonham. The control of discrete event systems. *Proceedings of the IEEE*, 77(1):81–98, January 1989.

[21] M. Sköldstam, K. Åkesson, and M. Fabian. Modeling of discrete event systems using finite automata with variables. In *Proceedings of 46th IEEE Conference on Decision and Control, CDC '07*, pages 3387–3392, December 2007.

[22] Rong Su, Jan H. van Schuppen, Jacobus E. Rooda, and Albert T. Hofkamp. Nonconflict check by using sequential automaton abstractions based on weak observation equivalence. *Automatica*, 46(6):968–978, June 2010.

[23] Y. Yang and R. Gohari. Embedded supervisory control of discrete-event systems. In *Proceedings of the 1st International Conference on Automation Science and Engineering, CASE 2005*, pages 410–415, Edmonton, Alberta, Canada, August 2005.

Appendix

This appendix contains the proofs of the propositions given in section 6. Most results about conflict equivalence of EFSM systems are proved by obtaining an unfolded FSM and using similar proofs about conflict equivalence of FSMs [9].

A Proof of Proposition 1

Prop. 1 concerns the relationship between synchronous composition of EFSMs and unfolding. In the following proof, it is shown that the results of unfolding before and after synchronous composition are identical up a renaming of the events.

Proposition 1 Let $\mathcal{E} = \{E_1, \ldots, E_n\}$ be an EFSM system, and $\mathcal{F} = \{E_1 \parallel E_2, E_3, \ldots, E_n\}$. Then $U(\mathcal{E})$ is nonblocking if and only if $U(\mathcal{F})$ is nonblocking.

Proof. It is shown in the following that the unfoldings $U(\mathcal{E})$ and $U(\mathcal{F})$ are identical up to a renaming of events, which is enough to show that $U(\mathcal{E})$ is nonblocking if and only if $U(\mathcal{F})$ is nonblocking. More precisely, it is shown that $U(\mathcal{F}) = \rho(U(\mathcal{E}))$ where $\rho\colon \Sigma_{\mathcal{E}} \to \Sigma_{\mathcal{F}}$ replaces events as follows,

$$\rho((E_i;\hat{v};\hat{w})) = \rho(E_i;\hat{v};\hat{w}) = \begin{cases} (E_1 \parallel E_2;\hat{v};\hat{w}), & \text{if } i = 1 \text{ or } i = 2; \\ (E_i;\hat{v};\hat{w}), & \text{otherwise}. \end{cases} \tag{15}$$

By definition 2, the EFSM systems $\mathcal{E}$ and $\mathcal{F}$ have the same variables associated with their update functions, so it holds that $\mathrm{vars}(\mathcal{E}) = \mathrm{vars}(\mathcal{F})$, and $\rho(U(\mathcal{E}))$ and $U(\mathcal{F})$ have the same states, initial states, and marked states. It remains to be shown that they also have the same transitions. Write $E = \|_{i=1}^{n} U(E_i)$ and $F = U(E_1 \parallel E_2) \parallel \|_{i=3}^{n} U(E_i)$.

First, let

$$(x_1, \ldots, x_n, \bar{v}) \xrightarrow{(E_k;\hat{v};\hat{w})} (y_1, \ldots, y_n, \bar{w}) \tag{16}$$

in $U(\mathcal{E})$. Then it follows that $(x_1, \ldots, x_n) \xrightarrow{(E_k;\hat{v};\hat{w})} (y_1, \ldots, y_n)$ in E, which means that $x_k \xrightarrow{(E_k;\hat{v};\hat{w})} y_k$ in $U(E_k)$ and $x_i = y_i$ for each $i \neq k$. It follows that there exists a transition $x_k \xrightarrow{p} y_k$ with $p(\hat{v}, \hat{w}) = \mathrm{true}$ in E_k. If $k = 1$ or $k = 2$, then either $(x_1, x_2) \xrightarrow{p} (y_1, x_2)$ or $(x_1, x_2) \xrightarrow{p} (x_1, y_2)$ in $E_1 \parallel E_2$ by definition 2, and thus in both cases $(x_1, \ldots, x_n) \xrightarrow{(E_1\|E_2;\hat{v};\hat{w})} (y_1, \ldots, y_n)$ in F. If $3 \leq k \leq n$, it follows directly from $x_k \xrightarrow{p} y_k$ that $(x_1, \ldots, x_n) \xrightarrow{(E_k;\hat{v};\hat{w})} (y_1, \ldots, y_n)$ in F. This shows that $(x_1, \ldots, x_n) \xrightarrow{\rho(E_k;\hat{v};\hat{w})} (y_1, \ldots, y_n)$ in F. From (16) it also follows that $\bar{v} \xrightarrow{(E_k;\hat{v};\hat{w})} \bar{w}$ in $\|_{v\in\mathrm{vars}(\mathcal{E})} U_{\mathcal{E}}(v)$ and thus $\bar{v} \xrightarrow{\rho(E_k;\hat{v};\hat{w})} \bar{w}$ in $\|_{v\in\mathrm{vars}(\mathcal{F})} U_{\mathcal{F}}(v)$ by definition 6. It follows that $(x_1, \ldots, x_n, \bar{v}) \xrightarrow{\rho(E_k;\hat{v};\hat{w})} (y_1, \ldots, y_n, \bar{w})$ in $U(\mathcal{F})$.

Conversely, let

$$(x_1, \ldots, x_n, \bar{v}) \xrightarrow{(G;\hat{v};\hat{w})} (y_1, \ldots, y_n, \bar{w}) \tag{17}$$

in $U(\mathcal{F})$, where $G \in \mathcal{F}$. Then $(x_1, \ldots, x_n) \xrightarrow{(G;\hat{v};\hat{w})} (y_1, \ldots, y_n)$ in F. Consider two cases.

- If $G = E_1 \parallel E_2$, then $(x_1, x_2) \xrightarrow{(E_1\|E_2;\hat{v};\hat{w})} (y_1, y_2)$ in $U(E_1 \parallel E_2)$ and $x_i = y_i$ for each $3 \le i \le n$. The former means $(x_1, x_2) \xrightarrow{p} (y_1, y_2)$ in $E_1 \parallel E_2$ with $p(\hat{v}, \hat{w}) = \mathrm{true}$, which by definition 2 implies $x_1 \xrightarrow{p} y_1$ in E_1 and $x_2 = y_2$, or $x_1 = y_1$ and $x_2 \xrightarrow{p} y_2$ in E_2. It follows that $(x_1, \ldots, x_n) \xrightarrow{(E_k;\hat{v};\hat{w})} (y_1, \ldots, y_n)$ in E, with $k = 1$ or $k = 2$, and thus $\rho(E_k; \hat{v}; \hat{w}) = (E_1 \parallel E_2; \hat{v}; \hat{w}) = (G; \hat{v}; \hat{w})$.

- If $G = E_k$ for some $3 \le k \le n$, then $x_k \xrightarrow{(E_k;\hat{v};\hat{w})} y_k$ in $U(E_k)$ and $(x_1, x_2) = (y_1, y_2)$ and $x_i = y_i$ for each $3 \le i \le n$ with $i \ne k$. This shows $(x_1, \ldots, x_n) \xrightarrow{(E_k;\hat{v};\hat{w})} (y_1, \ldots, y_n)$ in E, and $\rho(E_k; \hat{v}; \hat{w}) = (E_k; \hat{v}; \hat{w}) = (G; \hat{v}; \hat{w})$.

From (17), it also follows that $\bar{v} \xrightarrow{(G;\hat{v};\hat{w})} \bar{w}$ in $\|_{v\in\mathrm{vars}(\mathcal{F})} U_{\mathcal{F}}(v)$, and therefore $\bar{v} \xrightarrow{(E_k;\hat{v};\hat{w})} \bar{w}$ in $\|_{v\in\mathrm{vars}(\mathcal{E})} U_{\mathcal{E}}(v)$ by definition 6, where $\rho(E_k; \hat{v}; \hat{w}) = (G; \hat{v}; \hat{w})$. Then $(x_1, \ldots, x_n, \bar{v}) \xrightarrow{(E_k;\hat{v};\hat{w})} (y_1, \ldots, y_n, \bar{w})$ in $U(\mathcal{E})$, and therefore $(x_1, \ldots, x_n, \bar{v}) \xrightarrow{\rho(E_k;\hat{v};\hat{w})} (y_1, \ldots, y_n, \bar{w})$ in $\rho(U(\mathcal{E}))$. □

B Proof of Proposition 2

As conflict equivalence is preserved under bisimulation, the key step to prove proposition 2 is to show that the result of partial unfolding is bisimilar to the original. This is done in lemma 8. Before that, lemma 7 shows that the nonblocking property of EFSM systems is preserved when replacing subsystems by conflict equivalent subsystems.

Lemma 7 Let $\mathcal{E} = \{E_1, \ldots, E_n\}$ and $\mathcal{F} = \{F_1, E_2, \ldots, E_n\}$ be EFSM systems such that $\mathrm{vars}(\mathcal{E}) \setminus \mathrm{vars}(E_1) = \mathrm{vars}(\mathcal{F}) \setminus \mathrm{vars}(F_1)$ and

$$(U(E_1) \parallel \mathop{\Big\|}_{v\,\in\,\mathrm{vars}(E_1)} U_{\mathcal{E}}(v)) \setminus \Sigma_{E_1} \simeq_{\mathrm{conf}} (U(F_1) \parallel \mathop{\Big\|}_{v\,\in\,\mathrm{vars}(F_1)} U_{\mathcal{F}}(v)) \setminus \Sigma_{F_1} \,. \tag{18}$$

Then $U(\mathcal{E})$ is nonblocking if and only if $U(\mathcal{F})$ is nonblocking.

Proof. Note that

$$\begin{aligned} U(\mathcal{E}) &= U(E_1) \parallel \cdots \parallel U(E_n) \parallel \mathop{\Big\Vert}_{v \,\in\, \mathrm{vars}(\mathcal{E})} U_{\mathcal{E}}(v) \\ &= U(E_1) \parallel \mathop{\Big\Vert}_{v \,\in\, \mathrm{vars}(E_1)} U_{\mathcal{E}}(v) \;\parallel\; U(E_2) \parallel \cdots \parallel U(E_n) \parallel \mathop{\Big\Vert}_{v \,\in\, \mathrm{vars}(\mathcal{E}) \,\setminus\, \mathrm{vars}(E_1)} U_{\mathcal{E}}(v)\,. \end{aligned} \tag{19}$$

As the events in Σ_{E_1} do not appear in $U(E_2) \parallel \cdots \parallel U(E_n) \parallel \Vert_{v \in \mathrm{vars}(\mathcal{E}) \setminus \mathrm{vars}(E_1)} U_{\mathcal{E}}(v)$, the above (19) is nonblocking if and only if

$$(U(E_1) \parallel \mathop{\Big\Vert}_{v \,\in\, \mathrm{vars}(E_1)} U_{\mathcal{E}}(v)) \setminus \Sigma_{E_1} \quad\parallel\quad U(E_2) \parallel \cdots \parallel U(E_n) \parallel \mathop{\Big\Vert}_{v \,\in\, \mathrm{vars}(\mathcal{E}) \,\setminus\, \mathrm{vars}(E_1)} U_{\mathcal{E}}(v) \tag{20}$$

is nonblocking. Given (18) and noting that $\mathrm{vars}(\mathcal{E}) \setminus \mathrm{vars}(E_1) = \mathrm{vars}(\mathcal{F}) \setminus \mathrm{vars}(F_1)$, it follows from the definition of conflict equivalence (definition 11) that (20) is nonblocking if and only if

$$(U(F_1) \parallel \mathop{\Big\Vert}_{v \,\in\, \mathrm{vars}(F_1)} U_{\mathcal{F}}(v)) \setminus \Sigma_{F_1} \quad\parallel\quad U(E_2) \parallel \cdots \parallel U(E_n) \parallel \mathop{\Big\Vert}_{v \,\in\, \mathrm{vars}(\mathcal{F}) \,\setminus\, \mathrm{vars}(F_1)} U_{\mathcal{F}}(v) \tag{21}$$

is nonblocking. As the events in Σ_{F_1} do not appear in the FSMs $U(E_2), \ldots, U(E_n)$, or $U_{\mathcal{F}}(v)$ with $v \in \mathrm{vars}(\mathcal{F}) \setminus \mathrm{vars}(F_1)$, the above (21) is nonblocking if and only if

$$\begin{aligned} &U(F_1) \parallel \mathop{\Big\Vert}_{v \,\in\, \mathrm{vars}(F_1)} U_{\mathcal{F}}(v)) \;\parallel\; U(E_2) \parallel \cdots \parallel U(E_n) \parallel \mathop{\Big\Vert}_{v \,\in\, \mathrm{vars}(\mathcal{F}) \,\setminus\, \mathrm{vars}(F_1)} U_{\mathcal{F}}(v) = \\ &\qquad U(F_1) \parallel U(E_2) \parallel \cdots \parallel U(E_n) \parallel \mathop{\Big\Vert}_{v \,\in\, \mathrm{vars}(\mathcal{F})} U_{\mathcal{F}}(v) = U(\mathcal{F}) \end{aligned} \tag{22}$$

is nonblocking. □

Instead of showing that conflict equivalence is preserved under partial unfolding, lemma 8 below shows that the unfolded EFSM systems before and after partial unfolding are bisimilar FSMs, according to the following definition 18. Further, definition 19 provides notation to relate the valuations before and after partial unfolding to each other.

Definition 18 Let $G = \langle \Sigma_G, Q_G, \rightarrow_G, Q_G^\circ, Q_G^\omega \rangle$ and $H = \langle \Sigma_H, Q_H, \rightarrow_H, Q_H^\circ, Q_H^\omega \rangle$ be two FSMs. A relation $\approx \subseteq Q_G \times Q_H$ is called a *bisimulation equivalence* relation between G and H if the following holds for all $x_G \in Q_G$ and $x_H \in Q_H$ such that $x_G \approx x_H$:

- if $x_G \xrightarrow{\sigma}_G y_G$ for some $\sigma \in \Sigma_\tau$, then there exists $y_H \in Q_H$ such that $x_H \xrightarrow{\sigma}_H y_H$ and $y_G \approx y_H$;
- if $x_H \xrightarrow{\sigma}_H y_H$ for some $\sigma \in \Sigma_\tau$, then there exists $y_G \in Q_G$ such that $x_G \xrightarrow{\sigma}_G y_G$ and $y_G \approx y_H$;
- $x_G \in Q_G^\omega$ if and only if $x_H \in Q_H^\omega$;

G and H are *bisimilar*, written $G \approx H$, if there exists a bisimulation equivalence relation $\approx$ between G and H such that, for each $x_G^\circ \in Q_G^\circ$ there exists $x_H^\circ \in Q_H^\circ$ such that $x_G^\circ \approx x_H^\circ$, and vice versa.

Definition 19 Let $\bar{v} \colon V \to D$ be a valuation. The *restriction* $\bar{v}_{|W} \colon W \to D$ of $\bar{v}$ to $W \subseteq V$ is defined by

$$\bar{v}_{|W}[v] = \bar{v}[v] \quad \text{for all } v \in W \ . \tag{23}$$

For a variable v_0 and $a_0 \in \mathrm{dom}(v_0)$, the *extension* $\bar{v} \oplus \{v_0 \mapsto a_0\} \colon V \cup \{v_0\} \to D \cup \{a_0\}$ is defined by

$$\bar{v} \oplus \{v_0 \mapsto a_0\}[v] = \begin{cases} a_0, & \text{if } v = v_0 \ ; \\ \bar{v}[v], & \text{otherwise} \ . \end{cases} \tag{24}$$

Lemma 8 Let $\mathcal{E} = \{E_1, \ldots, E_n\}$ be an EFSM system, let $z \in \mathrm{vars}(E_1) \setminus \bigcup_{i=2}^n \mathrm{vars}(E_i)$, and let $\mathcal{F} = \{E_1 \setminus z, E_2, \ldots, E_n\}$. Then

$$(U(E_1) \parallel \mathop{\Big\|}_{v \,\in\, \mathrm{vars}(E_1)} U_{\mathcal{E}}(v)) \setminus \Sigma_{E_1} \ \approx \ (U(E_1 \setminus z) \parallel \mathop{\Big\|}_{v \,\in\, \mathrm{vars}(E_1 \,\setminus\, z)} U_{\mathcal{F}}(v)) \setminus \Sigma_{E_1 \setminus z} \ . \tag{25}$$

Proof. Let $F_1 = E_1 \setminus z$ and $F_i = E_i$ for $2 \le i \le n$. Write $E_1 = \langle V, Q, \to, Q^\circ, Q^\omega \rangle$, $E = U(E_1) \parallel \big\|_{v \in \mathrm{vars}(E_1)} U_{\mathcal{E}}(v)$ and $F = U(E_1 \setminus z) \parallel \big\|_{v \in \mathrm{vars}(F_1)} U_{\mathcal{F}}(v)$, and $\Sigma_E = \Sigma_{E_1}$ and $\Sigma_F = \Sigma_{F_1}$, and $W = \mathrm{vars}(E_1 \setminus z) = \mathrm{vars}(E_1) \setminus \{z\}$. The states of E have the form $(x, \bar{v})$, and the states of F have the form $((x, a), \check{v})$, where $x \in Q$, $\bar{v} \in \mathrm{dom}(\mathrm{vars}(E_1))$, $\check{v} \in \mathrm{dom}(W)$, and $a \in \mathrm{dom}(z)$. Consider the relation $\approx_U$ between the states of E and F, defined by

$$(x, \bar{v}) \approx_U ((y, a), \check{v}) \quad \text{if and only if} \quad x = y \text{ and } \bar{v} = \check{v} \oplus \{z \mapsto a\} \ . \tag{26}$$

It is to be shown that $\approx_U$ is a bisimulation between $E \setminus \Sigma_E$ and $F \setminus \Sigma_F$.

First, let $(x, \bar{v}) \approx_U ((x, a), \check{v})$ and $(x, \bar{v}) \xrightarrow{\sigma} (y, \bar{w})$ in $E \setminus \Sigma_E$. The former implies $\bar{v} = \check{v} \oplus \{z \mapsto a\}$ and thus $\bar{v}[z] = a$. Let $b = \bar{w}[z]$ and $\check{w} = \bar{w}_{|W}$. Also, as $(x, \bar{v}) \xrightarrow{\sigma} (y, \bar{w})$ in $E \setminus \Sigma_E$, there exists a transition $(x, \bar{v}) \xrightarrow{(E_k; \hat{v}; \hat{w})} (y, \bar{w})$ in E. Consider two cases.

- If $k = 1$, then $(E_k; \hat{v}; \hat{w}) = (E_1; \hat{v}; \hat{w}) \in \Sigma_E$ and thus $\sigma = \tau$. Since $x \xrightarrow{(E_1; \hat{v}; \hat{w})} y$ in $U(E_1)$, it holds that $x \xrightarrow{p} y$ in E_1 such that $p(\hat{v}, \hat{w}) = \mathrm{true}$. If $z \in \mathrm{vars}(p)$ then $z \in \mathrm{vars}(\hat{v})$ and $\hat{v}[z] = \bar{v}[z] = a$, and if $z \in \mathrm{vars}'(p)$ then $z \in \mathrm{vars}(\hat{w})$ and $\hat{w}[z] = \bar{w}[z] = b$. It follows that $(\exists z \exists z'(p \land z = a \land z' = b))(\hat{v}_{|W}, \hat{w}_{|W}) = \mathrm{true}$. By definition 10 there exists a transition $(x, a) \xrightarrow{\exists z \exists z'(p \land z = a \land z' = b)} (y, b)$ in $E_1 \setminus z$, which implies $(x, a) \xrightarrow{(E_1 \setminus z; \hat{v}_{|W}; \hat{w}_{|W})} (y, b)$ in $U(E_1 \setminus z)$.
- If $2 \le k \le n$, then the event $\sigma = (E_k; \hat{v}; \hat{w})$ is not in the alphabet of $U(E_1)$ or $U(E_1 \setminus z)$. It follows from $x \xrightarrow{(E_k; \hat{v}; \hat{w})} y$ in $U(E_1)$ that $x = y$. Since z is a local variable to E_1, it

does not appear in E_k with $2 \leq k \leq n$, so by definition 6 the event $(E_k; \hat{v}; \hat{w})$ is not in the alphabet of $U_{\mathcal{E}}(z)$. As $a = \bar{v}[z] \xrightarrow{(E_k;\hat{v};\hat{w})} \bar{w}[z] = b$ in $U_{\mathcal{E}}(z)$, it follows that $a = b$. Also $z \notin \text{vars}(\hat{v}) \cup \text{vars}(\hat{w})$ by definition 5, and thus $(E_k; \hat{v}; \hat{w}) = (E_k; \hat{v}_{|W}; \hat{w}_{|W})$. It follows that $(x, a) \xrightarrow{(E_k;\hat{v}_{|W};\hat{w}_{|W})} (x, a) = (y, b)$ in $U(E_1 \setminus z)$.

In both cases, it has been shown that $(x, a) \xrightarrow{(F_k;\hat{v}_{|W};\hat{w}_{|W})} (y, b)$ in $U(E_1 \setminus z)$ where $(F_k; \hat{v}_{|W}; \hat{w}_{|W}) = (E_1 \setminus z; \hat{v}_{|W}; \hat{w}_{|W})$ or $(F_k; \hat{v}_{|W}; \hat{w}_{|W}) = (E_k; \hat{v}_{|W}; \hat{w}_{|W})$. Further, $\bar{v} \xrightarrow{(E_k;\hat{v};\hat{w})} \bar{w}$ in $\|_{v \in \text{vars}(E_1)} U_{\mathcal{E}}(v)$ and thus it holds that $\check{v} \xrightarrow{(E_k;\hat{v};\hat{w})} \check{w}$ in $\|_{v \in \text{vars}(E_1) \setminus \{z\}} U_{\mathcal{E}}(v)$, which implies $\check{v} \xrightarrow{(F_k;\hat{v}_{|W};\hat{w}_{|W})} \check{w}$ in $\|_{v \in \text{vars}(E_1 \setminus z)} U_{\mathcal{F}}(v)$. This shows that $((x, a), \check{v}) \xrightarrow{(F_k;\hat{v}_{|W};\hat{w}_{|W})} ((y, b), \check{w})$ in F, and therefore $((x, a), \check{v}) \xrightarrow{\sigma} ((y, b), \check{w})$ in $F \setminus \Sigma_F$, with $(y, \bar{w}) \approx_U ((y, b), \check{w})$.

Conversely, assume that $(x, \bar{v}) \approx_U ((x, a), \check{v})$ and $((x, a), \check{v}) \xrightarrow{\sigma} ((y, b), \check{w})$ in $F \setminus \Sigma_F$. The former implies $\bar{v} = \check{v} \oplus \{z \mapsto a\}$ and thus $\bar{v}[z] = a$, and the latter implies the existence of a transition $((x, a), \check{v}) \xrightarrow{(F_k;\hat{v};\hat{w})} ((y, b), \check{w})$ in F. Consider two cases.

- If $k = 1$, then $(F_k; \hat{v}; \hat{w}) = (E_1 \setminus z; \hat{v}; \hat{w}) \in \Sigma_F$ and thus $\sigma = \tau$. Since $x \xrightarrow{(E_1 \setminus z;\hat{v};\hat{w})} y$ in $U(E_1 \setminus z)$, there is a transition $(x, a) \xrightarrow{\exists z \exists z'(p \wedge z = a \wedge z' = b)} (y, b)$ in $E_1 \setminus z$ with $x \xrightarrow{p} y$ in E_1, and $(\exists z \exists z'(p \wedge z = a \wedge z' = b))(\hat{v}, \hat{w}) = \text{true}$, and if $z \notin \text{vars}'(p)$ then $a = b$. Let

$$\hat{\hat{v}} = \begin{cases} \hat{v} \oplus \{z \mapsto a\}, & \text{if } z \in \text{vars}(p); \\ \hat{v}, & \text{otherwise}; \end{cases} \qquad \hat{\hat{w}} = \begin{cases} \hat{w} \oplus \{z \mapsto b\}, & \text{if } z \in \text{vars}'(p); \\ \hat{w}, & \text{otherwise}. \end{cases} \tag{27}$$

  Then it follows that $p(\hat{\hat{v}}, \hat{\hat{w}}) = \text{true}$, and therefore $x_1 \xrightarrow{(E_1,\hat{\hat{v}},\hat{\hat{w}})} y_1$ in $U(E_1)$.

  If $z \in \text{vars}(\hat{v}) \setminus \text{vars}(\hat{w})$, then $z \in \text{vars}(p) \setminus \text{vars}'(p)$, and $\hat{\hat{v}}[z] = a$ by construction (27), and $a = b$ as $z \notin \text{vars}'(p)$; it follows that $a \xrightarrow{(E_1,\hat{\hat{v}},\hat{\hat{w}})} a = b$ in $U_{\mathcal{E}}(z)$ by definition 6. If $z \in \text{vars}(\hat{w})$, then $z \in \text{vars}'(p)$, and $\hat{\hat{v}}[z] = a$ and $\hat{\hat{w}}[z] = b$ by construction (27); it follows that $a \xrightarrow{(E_1,\hat{\hat{v}},\hat{\hat{w}})} b$ in $U_{\mathcal{E}}(z)$ by definition 6. Otherwise $z \notin \text{vars}(\hat{v}) = \text{vars}(p) \supseteq \text{vars}'(p)$ in which case the event $(E_k, \hat{\hat{v}}, \hat{\hat{w}}) = (E_k; \hat{v}; \hat{w})$ is not in the alphabet of $U_{\mathcal{E}}(z)$, and $a = b$ as $z \notin \text{vars}'(p)$; it again follows that $a \xrightarrow{(E_1,\hat{\hat{v}},\hat{\hat{w}})} a = b$ in $U_{\mathcal{E}}(z)$.

- If $2 \leq k \leq n$, then the event $\sigma = (F_k; \hat{v}; \hat{w}) = (E_k; \hat{v}; \hat{w})$ is not in the alphabet of E_1 or $U(E_1)$. Then let $\hat{\hat{v}} = \hat{v}$ and $\hat{\hat{w}} = \hat{w}$ and $b = a$. It follows from $x \xrightarrow{(E_k;\hat{\hat{v}};\hat{\hat{w}})} y$ in $U(E_1)$ that $x = y$, and thus $x \xrightarrow{(E_k,\hat{\hat{v}},\hat{\hat{w}})} x = y$ in $U(E_1)$.

  Furthermore, since z is local to E_1, it does not appear in E_k with $2 \leq k \leq n$, and thus $z \notin \text{vars}(\hat{v}) \cup \text{vars}(\hat{w}) = \text{vars}(\hat{\hat{v}}) \cup \text{vars}(\hat{\hat{w}})$. It follows that $a \xrightarrow{(E_k,\hat{\hat{v}},\hat{\hat{w}})} a = b$ in $U_{\mathcal{E}}(z)$.

Let $\bar{w} = \check{w} \oplus \{z \mapsto b\}$. In both cases, it has been shown that $(x,a) \xrightarrow{(E_1,\hat{\hat{v}},\hat{\hat{w}})} (y,b)$ in $U(E_k) \parallel U_{\mathcal{E}}(z)$, with $\hat{v} \leq \hat{\hat{v}} \leq \bar{v}$ and $\hat{w} \leq \hat{\hat{w}} \leq \bar{w}$. Furthermore, note that $\check{v} \xrightarrow{(F_k;\hat{v};\hat{w})} \check{w}$ in $\|_{v \in \mathrm{vars}(E_1 \setminus z)} U_{\mathcal{F}}(v)$, which implies $\check{v} \xrightarrow{(E_k,\hat{\hat{v}},\hat{\hat{w}})} \check{w}$ in $\|_{v \in \mathrm{vars}(E_1) \setminus \{z\}} U_{\mathcal{E}}(v)$. Then it follows that $(x,\bar{v}) \xrightarrow{(E_k,\hat{\hat{v}},\hat{\hat{w}})} (y,\bar{w})$ in E, and hence $(x,\bar{v}) \xrightarrow{\sigma} (y,\bar{w})$ in $E \setminus \Sigma_E$, with $(y,\bar{w}) \approx_U ((y,b),\check{w})$.

As the FSMs E and F by construction have got exactly the same initial and marked states, it follows that $E \setminus \Sigma_E \approx F \setminus \Sigma_F$. □

Proposition 2 Let $\mathcal{E} = \{E_1, \ldots, E_n\}$ be an EFSM system, $z \in \mathrm{vars}(E_1) \setminus \bigcup_{i=2}^{n} \mathrm{vars}(E_i)$, and $\mathcal{F} = \{E_1 \setminus z, E_2, \ldots, E_n\}$. Then $U(\mathcal{E})$ is nonblocking if and only if $U(\mathcal{F})$ is nonblocking.

Proof. By lemma 8, it holds that

$$(U(E_1) \parallel \mathop{\Big\|}_{v \,\in\, \mathrm{vars}(E_1)} U_{\mathcal{E}}(v)) \setminus \Sigma_{E_1} \;\approx\; (U(E_1 \setminus z) \parallel \mathop{\Big\|}_{v \,\in\, \mathrm{vars}(E_1 \,\setminus\, z)} U_{\mathcal{F}}(v)) \setminus \Sigma_{E_1 \setminus z} \,. \tag{28}$$

As bisimulation of ordinary FSMs implies conflict equivalence [9], it follows that the above FSMs (28) are conflict equivalent. Furthermore, note that $\mathrm{vars}(\mathcal{E}) \setminus \mathrm{vars}(E_1) = \mathrm{vars}(\mathcal{F}) \setminus \mathrm{vars}(E_1 \setminus z)$ as the variable z is local to E_1 and does not appear in any EFSM E_i with $2 \leq i \leq n$. Then it follows from lemma 7 that $U(\mathcal{E})$ is nonblocking if and only if $U(\mathcal{F})$ is nonblocking. □

C Proof of Proposition 3

Before proving the key result about conflict equivalence in proposition 3, the following lemma 10 establishes a relationship between conflict equivalence of EFSMs and unfolded FSMs.

Definition 20 Let $G = \langle \Sigma_G, Q, \rightarrow, Q^{\circ}, Q^{\omega} \rangle$ be an FSM and $\Upsilon \subseteq \Sigma_G$. The result of *hiding* Υ from G, written $G \setminus \Upsilon$, is the FSM obtained from G by replacing each transition $x \xrightarrow{\sigma} y$ such that $\sigma \in \Upsilon$ by $x \xrightarrow{\tau} y$, and removing all events in Υ from Σ_G.

Lemma 10 Two EFSMs E_1 and F_1 are conflict equivalent, if and only if the following holds for all EFSM systems $\mathcal{E} = \{E_1, \ldots, E_n\}$ and $\mathcal{F} = \{F_1, E_2, \ldots, E_n\}$:

$$(U(E_1) \parallel \mathop{\Big\|}_{v \,\in\, \mathrm{vars}(E_1)} U_{\mathcal{E}}(v)) \setminus \Sigma_{E_1} \;\simeq_{\mathrm{conf}}\; (U(F_1) \parallel \mathop{\Big\|}_{v \,\in\, \mathrm{vars}(F_1)} U_{\mathcal{F}}(v)) \setminus \Sigma_{F_1} \,. \tag{29}$$

Proof. Assume that $E_1 \simeq_{\mathrm{conf}} F_1$. Furthermore, let $\mathcal{E} = \{E_1, \ldots, E_n\}$ and $\mathcal{F} = \{F_1, E_2, \ldots, E_n\}$ with $E_1, F_1 \notin \{E_2, \ldots, E_n\}$, and let $T = \langle \Sigma_T, Q_T, \rightarrow_T, Q_T^{\circ}, Q_T^{\omega} \rangle$ be an FSM such that

$$\big((U(E_1) \parallel \mathop{\Big\|}_{v \,\in\, \mathrm{vars}(E_1)} U_{\mathcal{E}}(v)\big) \setminus \Sigma_{E_1}) \parallel T \tag{30}$$

is nonblocking. Let $\Upsilon = \Sigma_T \setminus (\Sigma_{E_2} \cup \cdots \cup \Sigma_{E_n})$, and construct an EFSM E_T such that $U(E_T) \setminus \Upsilon = T \setminus \Upsilon$: this EFSM can be constructed as $E_T = \langle \mathrm{vars}(E_1), Q_T, \rightarrow_E, Q_T^\circ, Q_T^\omega \rangle$ where $x \xrightarrow{v=\hat{v} \wedge v'=\hat{w}}_E y$ for all transitions $x \xrightarrow{(E_i;\hat{v};\hat{w})}_T y$ with $2 \leq i \leq n$ and $x \xrightarrow{\mathrm{true}}_E y$ for all transitions $x \xrightarrow{\sigma}_T y$ with $\sigma \in \Upsilon$. Then

$$\begin{aligned} U(\{E_1, E_T\}) \setminus \Upsilon &= \bigl(U(E_1) \parallel U(E_T) \parallel \mathop{\parallel}_{v \,\in\, \mathrm{vars}(E_1)} U_{\mathcal{E}}(v))\bigr) \setminus \Upsilon \\ &= \bigl(U(E_1) \parallel \mathop{\parallel}_{v \,\in\, \mathrm{vars}(E_1)} U_{\mathcal{E}}(v))\bigr) \parallel (U(E_T) \setminus \Upsilon) \\ &= \bigl(U(E_1) \parallel \mathop{\parallel}_{v \,\in\, \mathrm{vars}(E_1)} U_{\mathcal{E}}(v))\bigr) \parallel (T \setminus \Upsilon) \end{aligned} \tag{31}$$

is nonblocking because (30) is nonblocking. Then $U(\{E_1, E_T\})$ is also nonblocking, and as $E_1 \simeq_{\mathrm{conf}} F_1$ it follows that $U(\{F_1, E_T\})$ is nonblocking. Then

$$\begin{aligned} U(\{F_1, E_T\}) \setminus \Upsilon &= \bigl(U(F_1) \parallel U(E_T) \parallel \mathop{\parallel}_{v \,\in\, \mathrm{vars}(F_1)} U_{\mathcal{E}}(v))\bigr) \setminus \Upsilon \\ &= \bigl(U(F_1) \parallel \mathop{\parallel}_{v \,\in\, \mathrm{vars}(F_1)} U_{\mathcal{E}}(v))\bigr) \parallel (U(E_T) \setminus \Upsilon) \\ &= \bigl(U(F_1) \parallel \mathop{\parallel}_{v \,\in\, \mathrm{vars}(F_1)} U_{\mathcal{E}}(v))\bigr) \parallel (T \setminus \Upsilon) \end{aligned} \tag{32}$$

is also nonblocking, and thus $((U(F_1) \parallel \parallel_{v \in \mathrm{vars}(F_1)} U_{\mathcal{E}}(v)) \setminus \Sigma_{F_1}) \parallel T$ is nonblocking. As T was chosen arbitrarily, it follows that the FSMs (29) are conflict equivalent.

Conversely assume that (29) holds, and let E_T be an EFSM such that $E_1 \parallel E_T$ is nonblocking, i.e., $U(\{E_1, E_T\})$ is nonblocking. Then $U(E_1) \parallel U(E_T) \parallel \parallel_{v \in \mathrm{vars}(E_1)} U_{\mathcal{E}}(v)$ is nonblocking, and as $U(E_T)$ does not use any events in Σ_{E_1}, it follows that $((U(E_1) \parallel \parallel \parallel_{v \in \mathrm{vars}(E_1)} U_{\mathcal{E}}(v)) \setminus \Sigma_{E_1}) \parallel U(E_T) = (U(E_1) \parallel U(E_T) \parallel \parallel_{v \in \mathrm{vars}(E_1)} U_{\mathcal{E}}(v)) \setminus \Sigma_{E_1}$ is nonblocking. Then by (29), it follows that $((U(F_1) \parallel \parallel_{v \in \mathrm{vars}(F_1)} U_{\mathcal{E}}(v)) \setminus \Sigma_{F_1}) \parallel U(E_T) = (U(F_1) \parallel U(E_T) \parallel \parallel_{v \in \mathrm{vars}(F_1)} U_{\mathcal{E}}(v)) \setminus \Sigma_{F_1}$ is nonblocking, and thus $F_1 \parallel E_T$ is nonblocking. As E_T was chosen arbitrarily, it follows that $E_1 \simeq_{\mathrm{conf}} F_1$. □

Proposition 3 Let $\mathcal{E} = \{E_1, \ldots, E_n\}$ and $\mathcal{F} = \{F_1, E_2, \ldots, E_n\}$ be EFSM systems such that $E_1 \simeq_{\mathrm{conf}} F_1$. Then $U(\mathcal{E})$ is nonblocking if and only if $U(\mathcal{F})$ is nonblocking.

Proof. As $E_1 \simeq_{\mathrm{conf}} F_1$, it follows by lemma 10 that (29) holds. Then the claim follows by lemma 7. □

D Proof of Proposition 4

To prove proposition 4, the key step is to show that the unfolded EFSMs are also observation equivalent. This is done below in lemma 13. Before that lemma 12 establishes an auxiliary result needed for lemma 13 and lemma 16.

Lemma 12 Let $\mathcal{E} = \{E_1, \dots, E_n\}$, and let $E = U(E_1) \parallel \|_{v \in \mathrm{vars}(E_1)} U_{\mathcal{E}}(v)$.

(i) If $(x, \bar{v}) \xrightarrow{(E_1;\hat{v};\hat{w})} (y, \bar{w})$ in E, then $x \overset{\bar{v},\bar{w}}{\Longrightarrow} y$ in E_1.

(ii) If $x \overset{\bar{v},\bar{w}}{\Longrightarrow} y$ in E_1, then $(x, \bar{v}) \overset{\varepsilon}{\Rightarrow} (y, \bar{w})$ in $E \setminus \Sigma_{E_1}$.

Proof.

(i) It follows from $(x, \bar{v}) \xrightarrow{(E_1;\hat{v};\hat{w})} (y, \bar{w})$ that $x \xrightarrow{(E_1;\hat{v};\hat{w})} y$ in $U(E_1)$ and $\bar{v} \xrightarrow{(E_1;\hat{v};\hat{w})} \bar{w}$ in $\|_{v \in \mathrm{vars}(E_1)} U_{\mathcal{E}}(v)$. The former implies by definition 5 that $x \xrightarrow{p} y$ with $p(\hat{v}, \hat{w}) = \mathrm{true}$ in E_1 with $\mathrm{vars}(\hat{v}) = \mathrm{vars}(p)$ and $\mathrm{vars}(\hat{w}) = \mathrm{vars}'(p)$, and the latter implies by definition 6 that $\hat{v} \leq \bar{v}$ and $\hat{w} \leq \bar{w}$ and $\bar{w}_{|\mathrm{vars}(E_1) \setminus \mathrm{vars}'(p)} = \bar{w}_{|\mathrm{vars}(E_1) \setminus \mathrm{vars}(\hat{w})} \leq \bar{v}$. Then it follows by definition 13 that $x \overset{\bar{v},\bar{w}}{\Longrightarrow} y$ in E_1.

(ii) It is first shown that if $x \overset{\bar{v}}{\Rightarrow} y$ in E_1, then $(x, \bar{v}) \overset{\varepsilon}{\Rightarrow} (y, \bar{v})$ in $E \setminus \Sigma_{E_1}$. By definition 13 it follows from $x \overset{\bar{v}}{\Rightarrow} y$ that $x = x_0 \xrightarrow{p_1} \cdots \xrightarrow{p_m} x_m = y$ in E_1, where $\mathrm{vars}'(p_j) = \emptyset$ and $p_j(\bar{v}) = \mathrm{true}$ for $1 \leq j \leq m$. By definition 5, this means $x = x_0 \xrightarrow{(E_1;\hat{v}_1;\emptyset)} \cdots \xrightarrow{(E_1;\hat{v}_m;\emptyset)} x_m = y$ in $U(E_1)$ with $\hat{v}_j \leq \bar{v}$ for $1 \leq j \leq m$. And by definition 6, for each $v \in \mathrm{vars}(E_1)$ such that the event $(E_1; \hat{v}_j; \emptyset)$ is in the alphabet of $U_{\mathcal{E}}(v)$, it holds that $v \in \mathrm{vars}(\hat{v}_j)$ and $\bar{v}[v] = \hat{v}_j[v] \xrightarrow{(E_1;\hat{v}_j;\emptyset)} \hat{v}_j[v] = \bar{v}[v]$. This means $\bar{v} \xrightarrow{(E_1;\hat{v}_1;\emptyset)} \cdots \xrightarrow{(E_1;\hat{v}_m;\emptyset)} \bar{v}$ in $\|_{v \in \mathrm{vars}(E_1)} U_{\mathcal{E}}(v)$. As furthermore $(E_1; \hat{v}_j; \emptyset) \in \Sigma_{E_1}$, this is enough to show $(x, \bar{v}) \overset{\varepsilon}{\Rightarrow} (y, \bar{v})$ in $E \setminus \Sigma_{E_1}$.

Now it is shown that the above implies that, if $x \overset{\bar{v},\bar{w}}{\Longrightarrow} y$ in E_1, then $(x, \bar{v}) \overset{\varepsilon}{\Rightarrow} (y, \bar{w})$ in $E \setminus \Sigma_{E_1}$. By definition 13 it follows from $x \overset{\bar{v},\bar{w}}{\Longrightarrow} y$ that $x \overset{\bar{v}}{\Rightarrow} x_1 \xrightarrow{p} y_1 \overset{\bar{w}}{\Rightarrow} y$ with $p(\bar{v}, \bar{w}) = \mathrm{true}$ and $\bar{w}_{|\mathrm{vars}(E) \setminus \mathrm{vars}'(p)} \leq \bar{v}$. By definition 5, this means $x_1 \xrightarrow{(E_1;\hat{v};\hat{w})} y_1$ in $U(E_1)$ with $\mathrm{vars}(\hat{v}) = \mathrm{vars}(p)$ and $\mathrm{vars}(\hat{w}) = \mathrm{vars}'(p)$ and $\hat{v} \leq \bar{v}$ and $\hat{w} \leq \bar{w}$. And by definition 6, for each $v \in \mathrm{vars}(E_1)$ such that the event $(E_1; \hat{v}; \hat{w})$ is in the alphabet of $U_{\mathcal{E}}(v)$, there are two possibilities: either $v \in \mathrm{vars}(\hat{v}) \setminus \mathrm{vars}(\hat{w}) = \mathrm{vars}(p) \setminus \mathrm{vars}'(p)$ and $\bar{v}[v] = \hat{v}[v] \xrightarrow{(E_1;\hat{v};\hat{w})} \hat{v}[v] = \bar{w}_{|\mathrm{vars}(E) \setminus \mathrm{vars}'(p)}[v] = \bar{w}[v]$, or $v \in \mathrm{vars}(\hat{w})$ and $\bar{v}[v] = \hat{v}[v] \xrightarrow{(E_1;\hat{v};\hat{w})} \hat{w}[v] = \bar{w}[v]$. Therefore, $(x, \bar{v}) \xrightarrow{(E_1;\hat{v};\hat{w})} (y, \bar{w})$ in E. Given the above result about $x \overset{\bar{v}}{\Rightarrow} x_1$ and $y_1 \overset{\bar{w}}{\Rightarrow} y$, and noting that $(E_1; \hat{v}; \hat{w}) \in \Sigma_{E_1}$, it follows that $(x, \bar{v}) \overset{\varepsilon}{\Rightarrow} (y, \bar{w})$ in $E \setminus \Sigma_{E_1}$. □

The following lemma 13 relates EFSM observation equivalence to observation equivalence of ordinary FSMs. As observation equivalence of FSMs implies conflict equivalence [9], this is enough to prove proposition 4. The proofs are based on the following definition of FSM observation equivalence.

Definition 21 Let $G = \langle \Sigma_G, Q, \to, Q^\circ, Q^\omega \rangle$ be an FSM. For $x, y \in Q$ and $s \in \Sigma^*$, the relation $x \overset{s}{\Rightarrow} y$ denotes the existence of a trace $t \in \Sigma_\tau^*$ such that $s = P(t)$ and $x \xrightarrow{t} y$. Here, $P \colon \Sigma_\tau^* \to \Sigma^*$ is the *natural projection* that removes all τ events from a trace $s \in \Sigma_\tau^*$.

In words, $x \xrightarrow{s} y$ denotes a path from state x to state y with *exactly* the events in s, while $x \overset{s}{\Rightarrow} y$ denotes a path with an arbitrary number of silent events τ shuffled with the events of s. The notation is applied to state sets, $X \overset{s}{\Rightarrow} y$, and to FSMs, $G \overset{s}{\Rightarrow} x$, analogously to $\to$.

Definition 22 Let $G = \langle \Sigma_G, Q_G, \to_G, Q_G^\circ, Q_G^\omega \rangle$ and $H = \langle \Sigma_H, Q_H, \to_H, Q_H^\circ, Q_H^\omega \rangle$ be two FSMs. A relation $\sim \subseteq Q_G \times Q_H$ is called an *observation equivalence* relation between G and H if the following holds for all $x_G \in Q_G$ and $x_H \in Q_H$ such that $x_G \sim x_H$:

- if $x_G \xrightarrow{\sigma}_G y_G$ for some $\sigma \in \Sigma_\tau$, then there exists $y_H \in Q_H$ such that $x_H \overset{P(\sigma)}{\Longrightarrow}_H y_H$ and $y_G \sim y_H$;
- if $x_H \xrightarrow{\sigma}_H y_H$ for some $\sigma \in \Sigma_\tau$, then there exists $y_G \in Q_G$ such that $x_G \overset{P(\sigma)}{\Longrightarrow}_G y_G$ and $y_G \sim y_H$;
- if $x_G \in Q_G^\omega$ then $x_H \overset{\varepsilon}{\Rightarrow}_H Q_H^\omega$;
- if $x_H \in Q_H^\omega$ then $x_G \overset{\varepsilon}{\Rightarrow}_G Q_G^\omega$.

G and H are *observation equivalent*, written $G \sim H$, if there exists an observation equivalence relation $\sim$ between G and H such that, for each $x_G^\circ \in Q_G^\circ$ there exists $x_H^\circ \in Q_H^\circ$ such that $x_G^\circ \sim x_H^\circ$, and vice versa.

Lemma 13 Let $\mathcal{E} = \{E_1, \ldots, E_n\}$ be an EFSM system and let F_1 be an EFSM such that $\mathrm{vars}(E_1) = \mathrm{vars}(F_1)$ and $E_1 \sim F_1$, and let $\mathcal{F} = \{F_1, E_2, \ldots, E_l\}$. Then

$$(U(E_1) \parallel \mathop{\big\|}_{v \,\in\, \mathrm{vars}(E_1)} U_{\mathcal{E}}(v)) \setminus \Sigma_{E_1} \;\sim\; (U(F_1) \parallel \mathop{\big\|}_{v \,\in\, \mathrm{vars}(F_1)} U_{\mathcal{F}}(v)) \setminus \Sigma_{F_1} \;. \tag{33}$$

Proof. Let $E = U(E_1) \parallel \|_{v \in \mathrm{vars}(E_1)} U_{\mathcal{E}}(v)$ and $F = U(F_1) \parallel \|_{v \in \mathrm{vars}(F_1)} U_{\mathcal{F}}(v)$, and $\Sigma_E = \Sigma_{E_1}$ and $\Sigma_F = \Sigma_{F_1}$. As $E_1 \sim F_1$, there exists an observation equivalence relation $\sim$ between E_1 and F_1. Consider the relation $\sim_U$ between the states of E and F, defined by

$$(x_E, \bar{v}_E) \sim_U (x_F, \bar{v}_F) \text{ if and only if } x_E \sim x_F \text{ and } \bar{v}_E = \bar{v}_F \;. \tag{34}$$

Note that $\bar{v}_E$ and $\bar{v}_F$ are defined over the same variables as $\mathrm{vars}(E_1) = \mathrm{vars}(F_1)$. It is to be shown that $\sim_U$ is an observation equivalence between $E \setminus \Sigma_E$ and $F \setminus \Sigma_F$.

Assume $(x_E, \bar{v}_E) \sim_U (x_F, \bar{v}_F)$, i.e., $x_E \sim x_F$ and $\bar{v}_E = \bar{v}_F = \bar{v}$.

Firstly, let $(x_E, \bar{v}) \xrightarrow{\zeta} (y_E, \bar{w})$ in $E \setminus \Sigma_E$, where $\zeta = \tau$ or $\zeta = (E_k; \hat{v}; \hat{w})$ with $2 \leq k \leq n$. It is to be shown that there exists a state y_F in F_1 such that $(x_F, \bar{v}) \overset{P(\zeta)}{\Longrightarrow} (y_F, \bar{w})$ in $F \setminus \Sigma_F$ and $y_E \sim y_F$. Consider two cases.

(i) $\zeta = \tau$. In this case, there exists $(E_1; \hat{v}; \hat{w}) \in \Sigma_E$ such that $(x_E, \bar{v}) \xrightarrow{(E_1;\hat{v};\hat{w})} (y_E, \bar{w})$ in E. By lemma 12 (i), it follows that $x_E \overset{\bar{v},\bar{w}}{\Longrightarrow} y_E$ in E_1. As $x_E \sim x_F$, there exists a state y_F in F_1 such that $x_F \overset{\bar{v},\bar{w}}{\Longrightarrow} y_F$ in F_1 and $y_E \sim y_F$. By lemma 12 (ii), it follows that $(x_F, \bar{v}) \overset{\varepsilon}{\Rightarrow} (y_F, \bar{w})$ in $F \setminus \Sigma_F$, with $P(\zeta) = \varepsilon$ and $y_E \sim y_F$.

(ii) $\zeta = (E_k; \hat{v}; \hat{w})$ with $2 \leq k \leq n$. In this case, ζ is not in the alphabet of $U(E_1)$ or $U(F_1)$, so let $y_F = x_E$. It follows from $x_E \xrightarrow{\zeta} y_E$ that $x_E = y_E$, and $x_F \xrightarrow{\zeta} x_F \sim x_E = y_F$ in $U(F_1)$. As also $\bar{v} \xrightarrow{\zeta} \bar{w}$ in $\|_{v \in \mathrm{vars}(E_1)} U_{\mathcal{E}}(v) = \|_{v \in \mathrm{vars}(F_1)} U_{\mathcal{F}}(v)$, this is enough to show $(x_F, \bar{v}) \overset{\zeta}{\Rightarrow} (y_F, \bar{w})$ in $F \setminus \Sigma_F$ with $y_E \sim y_F$.

Secondly, let $(x_E, \bar{v}) \in Q^{\omega}_{E \setminus \Sigma_E}$. It is to be shown that $(x_F, \bar{v}) \overset{\varepsilon}{\Rightarrow} Q^{\omega}_{F \setminus \Sigma_F}$. Clearly, $x_E \in Q^{\omega}_{E_1}$ and thus $x_E \overset{\bar{v}}{\Rightarrow} x_E \in Q^{\omega}_{E_1}$ in E_1 by definition 13. As $x_E \sim x_F$, it follows hat $x_F \overset{\bar{v}}{\Rightarrow} x^{\omega}_F \in Q^{\omega}_{F_1}$ for some state y_F of F_1, which by lemma 12 (ii) implies $(x_F, \bar{v}) \overset{\varepsilon}{\Rightarrow} (x^{\omega}_F, \bar{v}) \in Q^{\omega}_{F_1} \times \mathrm{dom}(\mathrm{vars}(F_1)) = Q^{\omega}_{F \setminus \Sigma_F}$.

Thirdly, assume $(x^{\circ}_E, \bar{v}^{\circ})$ is an initial state of $E \setminus \Sigma_E$. It is to be shown that $Q^{\circ}_{F \setminus \Sigma_F} \overset{\varepsilon}{\Rightarrow} (x_F, \bar{v}^{\circ})$ for some state x_F of F_1. Clearly, $x^{\circ}_E \in Q^{\circ}_{E_1}$, and $x^{\circ}_E \overset{\bar{v}^{\circ}}{\Rightarrow} x^{\circ}_E$ in E_1 by definition 13. As $\sim$ is an observation equivalence relation between E_1 and F_1, there exists a state x_F of F_1 such that $Q^{\circ}_{F_1} \overset{\bar{v}^{\circ}}{\Rightarrow} x_F$ in F_1. That is, $x^{\circ}_F \overset{\bar{v}^{\circ}}{\Rightarrow} x_F$ for some $x^{\circ}_F \in Q^{\circ}_{F_1}$. By lemma 12 (ii), it follows that $(x^{\circ}_F, \bar{v}^{\circ}) \overset{\varepsilon}{\Rightarrow} (x_F, \bar{v}^{\circ})$ in $F \setminus \Sigma_F$, where $(x^{\circ}_F, \bar{v}^{\circ}) \in Q^{\circ}_{F \setminus \Sigma_F}$. □

Proposition 4 Let E_1 and F_1 be two EFSMs. If $E_1 \sim F_1$ then $E_1 \simeq_{\mathrm{conf}} F_1$.

Proof. By lemma 13, it holds that

$$(U(E_1) \parallel \mathop{\Big\|}_{v \,\in\, \mathrm{vars}(E_1)} U_{\mathcal{E}}(v)) \setminus \Sigma_{E_1} \;\sim\; (U(F_1) \parallel \mathop{\Big\|}_{v \,\in\, \mathrm{vars}(F_1)} U_{\mathcal{F}}(v)) \setminus \Sigma_{F_1}\,. \tag{35}$$

As observation equivalence of ordinary FSMs implies conflict equivalence [9], it follows that the above FSMs (35) are conflict equivalent. Then it follows from lemma 10 that $E_1 \simeq_{\mathrm{conf}} F_1$. □

E Proof of Proposition 5

Before proving that conflict equivalence is preserved by the Active Events Rule, lemma 15 shows that every path in the unfolded FSM of an EFSM also occurs in the unfolded FSM of every abstraction obtained by FSM quotient. Furthermore, lemma 16 guarantees that under the additional assumption of incoming equivalence, a converse of lemma 15 also holds.

Lemma 15 Let $\mathcal{E} = \{E_1, \dots, E_n\}$, let $\sim$ be an equivalence relation on the location set of E_1, and let $\mathcal{F} = \{E_1/{\sim}, E_2, \dots, E_n\}$. If $(x, \bar{v}) \overset{s}{\Rightarrow} (y, \bar{w})$ in $(U(E_1) \parallel \|_{v \in \mathrm{vars}(E_1)} U_{\mathcal{E}}(v)) \setminus \Sigma_{E_1}$ for some $s \in (\Sigma_{\mathcal{E}} \setminus \Sigma_{E_1})^*$, then $([x], \bar{v}) \overset{s}{\Rightarrow} ([y], \bar{w})$ in $(U(E_1/{\sim}) \parallel \|_{v \in \mathrm{vars}(E_1)} U_{\mathcal{F}}(v)) \setminus \Sigma_{E_1/\sim}$.

Proof.

Write $E = U(E_1) \parallel \|_{v \in \mathrm{vars}(E_1)} U_{\mathcal{E}}(v)$ and $F = U(E_1/{\sim}) \parallel \|_{v \in \mathrm{vars}(E_1/\sim)} U_{\mathcal{F}}(v)$, and let $s \in (\Sigma_{\mathcal{E}} \setminus \Sigma_{E_1})^*$ such that $(x, \bar{v}) \overset{s}{\Rightarrow} (y, \bar{w})$ in $E \setminus \Sigma_{E_1}$. Then there exists $s' = (F_1; \hat{v}_1; \hat{w}_1) \cdots (F_m; \hat{v}_m; \hat{w}_m) \in \Sigma_{\mathcal{E}}^*$ such that $P(s') = s$ and

$$(x, \bar{v}) = (x_0, \bar{v}_0) \xrightarrow{(F_1;\hat{v}_1;\hat{w}_1)} (x_1, \bar{v}_1) \xrightarrow{(F_2;\hat{v}_2;\hat{w}_2)} \cdots \xrightarrow{(F_m;\hat{v}_m;\hat{w}_m)} (x_m, \bar{v}_m) = (y, \bar{w}) \tag{36}$$

in E. Here, the natural projection $P \colon \Sigma^* \to (\Sigma_{\mathcal{E}} \setminus \Sigma_{E_1})^*$ erases events in Σ_{E_1} and $\Sigma_{E_1/\sim}$ from traces.

Consider a transition $(x_{i-1}, \bar{v}_{i-1}) \xrightarrow{(F_i;\hat{v}_i;\hat{w}_i)} (x_i, \bar{v}_i)$ on the path (36). If $F_i \neq E_1$, then the event $(F_i; \hat{v}_i; \hat{w}_i)$ is neither in the alphabet of $U(E_1)$ nor of $U(E_1/{\sim})$, and given $\mathrm{vars}(E_1) = \mathrm{vars}(E_1/{\sim})$ it follows immediately that the transition $([x_{i-1}], \bar{v}_{i-1}) \xrightarrow{(F_i;\hat{v}_i;\hat{w}_i)} ([x_i], \bar{v}_i)$ is in F. Otherwise $F_i = E_1$, which means $(x_{i-1}, \bar{v}_{i-1}) \xrightarrow{(E_1;\hat{v}_i;\hat{w}_i)} (x_i, \bar{v}_i)$. By definition 5 it holds that $x_{i-1} \xrightarrow{p} x_i$ in E_1 with $p(\bar{v}_i, \bar{w}_i) = \mathrm{true}$. This implies $[x_{i-1}] \xrightarrow{p} [x_i]$ in $E_1/{\sim}$, and $[x_{i-1}] \xrightarrow{(E_1/\sim;\hat{v}_i;\hat{w}_i)} [x_i]$ in $U(E_1/{\sim})$ by definition 5. Since $\bar{v}_{i-1} \xrightarrow{(E_1;\hat{v}_i;\hat{w}_i)} \bar{v}_i$ in $\|_{v \in \mathrm{vars}(E_1)} U_{\mathcal{E}}(v)$ and $\mathrm{vars}(E_1) = \mathrm{vars}(E_1/{\sim})$, thus $\bar{v}_{i-1} \xrightarrow{(E_1/\sim;\hat{v}_i;\hat{w}_i)} \bar{v}_i$ in $\|_{v \in \mathrm{vars}(E_1/\sim)} U_{\mathcal{F}}(v)$. This implies that $([x_{i-1}], \bar{v}_{i-1}) \xrightarrow{(E_1/\sim;\hat{v}_i;\hat{w}_i)} ([x_i], \bar{v}_i)$ in F. As this has been shown for all $1 \leq i \leq n$, the path

$$([x], \bar{v}) = ([x_0], \bar{v}_0) \xrightarrow{(F'_1;\hat{v}_1;\hat{w}_1)} ([x_1], \bar{v}_1) \xrightarrow{(F'_2;\hat{v}_2;\hat{w}_2)} \cdots \xrightarrow{(F'_m;\hat{v}_m;\hat{w}_m)} ([x_m], \bar{v}_m) = ([y], \bar{w}) \tag{37}$$

is in F, where $F'_i = E_1/{\sim}$ or $F'_i = E_k$ for some $2 \leq k \leq n$. It follows that $([x], \bar{v}) \overset{s}{\Rightarrow} ([y], \bar{w})$ in $F \setminus \Sigma_{E_1/\sim}$. □

Lemma 16 Let $\mathcal{E} = \{E_1, \dots, E_n\}$, let $\sim$ be an equivalence relation on the location set of E_1 such that $\sim \subseteq \sim_{\mathrm{inc}}$, and let $\mathcal{F} = \{E_1/{\sim}, \dots, E_2, E_n\}$. If $(\tilde{x}, \bar{v}) \overset{s}{\Rightarrow} (\tilde{y}, \bar{w})$ in $(U(E_1/{\sim}) \parallel \|_{v \in \mathrm{vars}(E_1)} U_{\mathcal{F}}(v)) \setminus$

$\Sigma_{E_1/\sim}$ for some $s \in (\Sigma_{\mathcal{E}} \setminus \Sigma_{E_1/\sim})^*$, then for all $y \in \tilde{y}$ there exists $x \in \tilde{x}$ such that $(x, \bar{v}) \overset{s}{\Rightarrow} (y, \bar{w})$ in $(U(E_1) \parallel \|_{v \in \mathrm{vars}(E_1)} U_{\mathcal{E}}(v)) \setminus \Sigma_{E_1}$.

Proof.

Write $E = U(E_1) \parallel \|_{v \in \mathrm{vars}(E_1)} U_{\mathcal{E}}(v)$ and $F = U(E_1/\sim) \parallel \|_{v \in \mathrm{vars}(E_1/\sim)} U_{\mathcal{F}}(v)$, and let $s \in (\Sigma_{\mathcal{E}} \setminus \Sigma_{E_1/\sim})^*$ such that $(\tilde{x}, \bar{v}) \overset{s}{\Rightarrow} (\tilde{y}, \bar{w})$ in $F \setminus \Sigma_{E_1/\sim}$. Then there exists $s' \in \Sigma_{\mathcal{F}}^*$ such that $P(s') = s$ and $(\tilde{x}, \bar{v}) \xrightarrow{s'} (\tilde{y}, \bar{w})$ in F. Without loss of generality, this path does not contain any selfloops labelled by events in $\Sigma_{E_1/\sim}$, and the natural projection $P\colon \Sigma^* \to (\Sigma_{\mathcal{E}} \setminus \Sigma_{E_1})^*$ erases events in Σ_{E_1} and $\Sigma_{E_1/\sim}$ from traces. Let $y \in \tilde{y}$. It is shown by induction on the length of s' that there exists $x \in \tilde{x}$ such that $(x, \bar{v}) \overset{P(s')}{\Longrightarrow} (y, \bar{w})$ in $E \setminus \Sigma_{E_1}$.

If $s' = \varepsilon$, this is clear with $x = y$ and $P(s') = \varepsilon$.

Now consider $s' = (F_0; \hat{v}; \hat{w})t$ such that $(\tilde{x}, \bar{v}) \xrightarrow{(F_0;\hat{v};\hat{w})} (\tilde{z}, \bar{v}') \xrightarrow{t} (\tilde{y}, \bar{w})$ in F, and assume by inductive assumption that there exists $z \in \tilde{z}$ such that $(z, \bar{v}') \overset{P(t)}{\Longrightarrow} (y, \bar{w})$ in $E \setminus \Sigma_{E_1}$. Consider two cases.

- If $F_0 = E_1/\sim$, then $(\tilde{x}, \bar{v}) \xrightarrow{(E_1/\sim;\hat{v};\hat{w})} (\tilde{z}, \bar{v}')$ in F. Then $\tilde{x} \xrightarrow{(E_1/\sim;\hat{v};\hat{w})} \tilde{z}$ in $U(E_1/\sim)$, so by definition 5 it holds that $\tilde{x} \xrightarrow{p} \tilde{z}$ in $E_1/\sim$ with $p(\hat{v}, \hat{w}) = \mathrm{true}$. It follows that there exist $x \in \tilde{x}$ and $z' \in \tilde{z}$ such that $x \xrightarrow{p} z'$ in E_1, which again by definition 5 means that $x \xrightarrow{(E_1;\hat{v};\hat{w})} z'$ in $U(E_1)$. As $\mathrm{vars}(E_1) = \mathrm{vars}(E_1/\sim)$, it follows that $(x, \bar{v}) \xrightarrow{(E_1;\hat{v};\hat{w})} (z', \bar{v}')$ in E. Then it follows by lemma 12 (i) that $x \overset{\bar{v},\bar{v}'}{\Longrightarrow} z'$ in E_1. As the path $(\tilde{x}, \bar{v}) \xrightarrow{s'} (\tilde{y}, \bar{w})$ does not contain any selfloops labelled by events in $\Sigma_{E_1/\sim}$, it holds that $\bar{v} \neq \bar{v}'$ or $\tilde{x} \neq \tilde{z}$, and the latter implies $x \neq z'$. Therefore, as $z \sim_{\mathrm{inc}} z'$, it follows that $x \overset{\bar{v},\bar{v}'}{\Longrightarrow} z$ in E_1, which by lemma 12 (ii) implies $(x, \bar{v}) \overset{\varepsilon}{\Rightarrow} (z, \bar{v}')$ in $E \setminus \Sigma_{E_1}$. This shows that $(z, \bar{v}) \overset{\varepsilon}{\Rightarrow} (z, \bar{v}') \overset{P(t)}{\Longrightarrow} (y, \bar{w})$ in $E \setminus \Sigma_{E_1}$, where $P(s) = P((E_1/\sim; \hat{v}; \hat{w})t) = P(t)$.
- If $F_0 \neq E_1/\sim$, i.e., $F_0 = E_i$ for some $2 \leq i \leq n$, then the event $(F_0; \hat{v}; \hat{w})$ is neither in the alphabet of $U(E_1/\sim)$ nor of $U(E_1)$, and it follows immediately that $z \in \tilde{z} = \tilde{x}$ and $z \xrightarrow{(F_0;\hat{v};\hat{w})} z$. As furthermore $\bar{v}' \xrightarrow{(F_0;\hat{v};\hat{w})} \bar{w}$ in $\|_{v \in \mathrm{vars}(E_1/\sim)} U_{\mathcal{F}}(v) = \|_{v \in \mathrm{vars}(E_1)} U_{\mathcal{E}}(v)$, it follows that $(z, \bar{v}) \xrightarrow{(F_0;\hat{v};\hat{w})} (z, \bar{v}')$ in E. This shows that $(z, \bar{v}) \xrightarrow{(F_0;\hat{v};\hat{w})} (z, \bar{v}') \overset{P(t)}{\Longrightarrow} (y, \bar{w})$ in $E \setminus \Sigma_{E_1}$, where $P(s) = P((F_0; \hat{v}; \hat{w})t) = (F_0; \hat{v}; \hat{w})P(t)$, so the claim follows with $x = z$. □

Proposition 5 Let $E_1 = \langle V, Q, \to, Q^\circ, Q^\omega \rangle$ be an EFSM, and let $\sim \subseteq Q \times Q$ be an equivalence relation such that $\sim \subseteq \sim_{\mathrm{inc}} \cap \sim_{\mathrm{act}}$, where $\sim_{\mathrm{inc}}$ and $\sim_{\mathrm{act}}$ are the incoming and active events equivalences of E_1. Then $E_1 \simeq_{\mathrm{conf}} E_1/\sim$.

Proof. Let $\mathcal{E} = \{E_1, \ldots, E_n\}$ and $\mathcal{F} = \{E_1/\sim, E_2, \ldots, E_n\}$. Furthermore, write $E = U(E_1) \parallel$

$\|_{v\in\mathrm{vars}(E_1)} U_{\mathcal{E}}(v)$ and $F = U(E_1/{\sim}) \parallel \|_{v\in\mathrm{vars}(E_1/{\sim})} U_{\mathcal{F}}(v)$, and $\Sigma_E = \Sigma_{E_1}$ and $\Sigma_F = \Sigma_{E_1/\sim}$. Using lemma 10, it is enough to show $E \setminus \Sigma_E \simeq_{\mathrm{conf}} F \setminus \Sigma_F$.

Let T such that $(E \setminus \Sigma_E) \parallel T$ is nonblocking, and assume $(F \setminus \Sigma_F) \parallel T \overset{s}{\Rightarrow} (\tilde{x}, \bar{v}, x_T)$. Then $F \setminus \Sigma_F \overset{s}{\Rightarrow} (\tilde{x}, \bar{v})$. Let $x \in \tilde{x}$. As $\sim \subseteq \sim_{\mathrm{inc}}$, it follows by lemma 16 that $E \setminus \Sigma_E \overset{s}{\Rightarrow} (x, \bar{v})$. Therefore, $(E \setminus \Sigma_E) \parallel T \overset{s}{\Rightarrow} (x, \bar{v}, x_T)$. As $(E \setminus \Sigma_E) \parallel T$ is nonblocking, there exists a trace t such that $(E \setminus \Sigma_E) \parallel T \overset{s}{\Rightarrow} (x, \bar{v}, x_T) \overset{t}{\Rightarrow} (x^\omega, \bar{w}, x_T^\omega)$ with $x^\omega \in Q_{E_1}^\omega$, $\bar{w} \in \mathrm{dom}(\mathrm{vars}(E_1))$, and $x_T^\omega \in Q_T^\omega$. Therefore, $(x, \bar{v}) \overset{t}{\Rightarrow} (x_E^\omega, \bar{w})$ in $E \setminus \Sigma_E$, and it follows by lemma 15 that $(\tilde{x}, \bar{v}) = ([x], \bar{v}) \overset{t}{\Rightarrow} ([x^\omega], \bar{w})$ in $F \setminus \Sigma_F$, where $[x^\omega] \in Q_{E_1/\sim}^\omega$ as $x^\omega \in Q_{E_1}^\omega$. Then it follows that

$$(F \setminus \Sigma_F) \parallel T \overset{s}{\Rightarrow} (\tilde{x}, \bar{v}, x_T) \overset{t}{\Rightarrow} ([x^\omega], \bar{w}, x_T^\omega) \in Q_{E_1/\sim}^\omega \times \mathrm{dom}(\mathrm{vars}(E_1/{\sim})) \times Q_T^\omega \tag{38}$$

which means $(F \setminus \Sigma_F) \parallel T$ is nonblocking.

Conversely, let T such that $(F \setminus \Sigma_F) \parallel T$ is nonblocking, and assume $(E \setminus \Sigma_E) \parallel T \overset{s}{\Rightarrow} (x, \bar{v}, x_T)$. Then $E \setminus \Sigma_E \overset{s}{\Rightarrow} (x, \bar{v})$, and it follows by lemma 15 that $F \setminus \Sigma_F \overset{s}{\Rightarrow} ([x], \bar{v})$. Therefore, $(F \setminus \Sigma_F) \parallel T \overset{s}{\Rightarrow} ([x], \bar{v}, x_T)$. As $(F \setminus \Sigma_F) \parallel T$ is nonblocking, there exists a trace t such that $(F \setminus \Sigma_F) \parallel T \overset{s}{\Rightarrow} ([x], \bar{v}, x_T) \overset{t}{\Rightarrow} (\tilde{x}^\omega, \bar{v}^\omega, x_T^\omega)$ with $\tilde{x}^\omega \in Q_{E_1/\sim}^\omega$, $\bar{v}^\omega \in \mathrm{dom}(\mathrm{vars}(E_1))$, and $x_T^\omega \in Q_T^\omega$. Assume without loss of generality that the path

$$([x], \bar{v}, x_T) \overset{t}{\Rightarrow} (\tilde{x}^\omega, \bar{v}^\omega, x_T^\omega) \tag{39}$$

does not contain any selfloops. As $\tilde{x}^\omega \in Q_{E_1/\sim}^\omega$, there exists $x^\omega \in \tilde{x}^\omega$ such that $x^\omega \in Q_{E_1}^\omega$. Also $([x], \bar{v}) \overset{t}{\Rightarrow} (\tilde{x}^\omega, \bar{v}^\omega)$ in $F \setminus \Sigma_F$, so by lemma 16, there exists $x' \in [x]$ such that $(x', \bar{v}) \overset{t}{\Rightarrow} (x^\omega, \bar{v}^\omega)$ in $E \setminus \Sigma_E$. Then there exists a trace t' such that $P(t') = t$ and $(x', \bar{v}) \xrightarrow{t'} (\tilde{x}^\omega, \bar{v}^\omega)$ in E, where the natural projection $P \colon \Sigma^* \to (\Sigma_{\mathcal{E}} \setminus \Sigma_{E_1})^*$ erases events in Σ_{E_1} and $\Sigma_{E_1/\sim}$ from traces. Let $p \sqsubseteq t'$ be the longest prefix of t' such that $p \in (\Sigma_{\mathcal{E}} \setminus \Sigma_E)^*$, so that $t' = pq$ with $p \in (\Sigma_{\mathcal{E}} \setminus \Sigma_E)^*$ and $q = \varepsilon$ or the first event of q is in Σ_E. Then $(x', \bar{v}) \xrightarrow{p} (x', \bar{v}') \xrightarrow{q} (x^\omega, \bar{v}^\omega)$ in E for some $\bar{v}' \in \mathrm{dom}(\mathrm{vars}(E_1))$ and $x_T \overset{p}{\Rightarrow} y_T \overset{q}{\Rightarrow} x_T^\omega$ for some state y_T of T. Consider two cases.

(i) If $q = \varepsilon$, then $x' = x^\omega \in Q_{E_1}^\omega$ and $y_T \overset{\varepsilon}{\Rightarrow} x_T^\omega$. This implies $x' \overset{\bar{v}'}{\Rightarrow} Q_{E_1}^\omega$ and thus $x \overset{\bar{v}'}{\Rightarrow} y^\omega \in Q_{E_1}^\omega$ for some y^ω since $x \sim_{\mathrm{act}} x'$. By lemma 12 (ii), it follows that $(x, \bar{v}') \overset{\varepsilon}{\Rightarrow} (y^\omega, \bar{v}')$ in $E \setminus \Sigma_E$. It follows that $(E \setminus \Sigma_E) \parallel T \overset{s}{\Rightarrow} (x, \bar{v}, x_T) \overset{p}{\Rightarrow} (x, \bar{v}', y_T) \overset{\varepsilon}{\Rightarrow} (y^\omega, \bar{v}', x_T^\omega) \in Q_{E_1}^\omega \times \mathrm{dom}(\mathrm{vars}(E_1)) \times Q_T^\omega$, i.e., $(E \setminus \Sigma_E) \parallel T$ is nonblocking.

(ii) If the first event of q is in Σ_E, then let $q = (E_1; \hat{v}; \hat{w})r$ and $(x', \bar{v}') \xrightarrow{(E_1;\hat{v};\hat{w})} (y, \bar{w}') \xrightarrow{r} (x^\omega, \bar{v}^\omega)$ in E. It follows from $(x', \bar{v}') \xrightarrow{(E_1;\hat{v};\hat{w})} (y, \bar{w}')$ by lemma 12 (i) that $x' \overset{\bar{v}',\bar{w}'}{\Longrightarrow} y$ in E_1. As the path (39) does not contain any selfloops, it holds that $\bar{v}' \neq \bar{w}'$ or $x' \neq y$. Then, since $x \sim_{\mathrm{act}} x'$, it follows that $x \overset{\bar{v}',\bar{w}'}{\Longrightarrow} y'$ in E_1 for some state y' of E_1, with $\bar{v}' \neq \bar{w}'$ or $x \neq y'$. This implies $(x, \bar{v}') \overset{\varepsilon}{\Rightarrow} (y', \bar{w}')$ in $E \setminus \Sigma_E$ by lemma 12 (ii), and thus $(E \setminus \Sigma_E) \parallel T \overset{s}{\Rightarrow} (x, \bar{v}, x_T) \overset{p}{\Rightarrow}$

$(x, \bar{v}', y_T) \overset{\varepsilon}{\Rightarrow} (y', \bar{w}', y_T)$. Then it follows by lemma 15 that $(F \setminus \Sigma_F) \parallel T \overset{sp}{\Rightarrow} ([x], \bar{v}', y_T) \overset{\varepsilon}{\Rightarrow} ([y'], \bar{w}', y_T)$. As $(F \setminus \Sigma_F) \parallel T$ is nonblocking, there exists a trace u such that $(F \setminus \Sigma_F) \parallel T \overset{sp}{\Rightarrow} ([y'], \bar{w}', y_T) \overset{u}{\Rightarrow} (\tilde{y}^\omega, \bar{w}^\omega, y_T^\omega)$ with $\tilde{y}^\omega \in Q^\omega_{E_1/\sim}$, $\bar{w}^\omega \in \mathrm{dom}(\mathrm{vars}(E_1))$, and $y_T^\omega \in Q_T^\omega$. As $\tilde{y}^\omega \in Q^\omega_{E_1/\sim}$, there exists $y^\omega \in \tilde{y}^\omega$ such that $y^\omega \in Q^\omega_{E_1}$. Also $([y'], \bar{w}') \overset{u}{\Rightarrow} (\tilde{y}^\omega, \bar{w}^\omega)$ in $F \setminus \Sigma_F$, so by lemma 16, there exists $y'' \in [y']$ such that $(y'', \bar{w}') \overset{u}{\Rightarrow} (y^\omega, \bar{w}^\omega)$ in $E \setminus \Sigma_E$. Now recall that $x \overset{\bar{v}', \bar{w}'}{\Longrightarrow} y'$ with $\bar{v}' \neq \bar{w}'$ or $x \neq y'$. Therefore, it follows from $y' \sim_{\mathrm{inc}} y''$ that $x \overset{\bar{v}', \bar{w}'}{\Longrightarrow} y''$, and thus $(x, \bar{v}') \overset{\varepsilon}{\Rightarrow} (y'', \bar{w}')$ in $E \setminus \Sigma_E$ by lemma 12 (ii). This implies $(E \setminus \Sigma_E) \parallel T \overset{s}{\Rightarrow} (x, \bar{v}, x_T) \overset{p}{\Rightarrow} (x, \bar{v}', y_T) \overset{\varepsilon}{\Rightarrow} (y'', \bar{w}', y_T) \overset{u}{\Rightarrow} (\tilde{y}^\omega, \bar{w}^\omega, y_T^\omega) \in Q^\omega_{E_1} \times \mathrm{dom}(\mathrm{vars}(E_1)) \times Q_T^\omega$, i.e., $(E \setminus \Sigma_E) \parallel T$ is nonblocking. □