Basic Research in Computer Science

Verification of State/Event Systems by Quotienting

Nicky O. Bodentien
Jacob Vestergaard
Jakob Friis
Kåre J. Kristoffersen
Kim G. Larsen

BRICS Report Series RS-99-41

ISSN 0909-0878 December 1999

See back inner page for a list of recent BRICS Report Series publications. Copies may be obtained by contacting:

BRICS
Department of Computer Science
University of Aarhus
Ny Munkegade, building 540
DK–8000 Aarhus C
Denmark
Telephone: +45 8942 3360
Telefax: +45 8942 3255
Internet: BRICS@brics.dk

BRICS publications are in general accessible through the World Wide Web and anonymous FTP through these URLs:

`http://www.brics.dk`
`ftp://ftp.brics.dk`
This document in subdirectory `RS/99/41/`

Verification of State/Event Systems by Quotienting

Nicky Oliver Bodentien *Jacob Vestergaard* *Jakob Friis*
Kåre Jelling Kristoffersen *Kim Guldstrand Larsen*

{oliver, jacob, freeze, jelling, kgl}@cs.auc.dk
Aalborg University

BRICS[*]
Department of Computer Science
Aalborg University
Fredrik Bajers Vej 7E
DK-9220 Aarhus Ø, Denmark

Abstract

A rather new approach towards compositional verification of concurrent systems is the quotient technique, where components are gradually removed from the concurrent system while transforming the specification accordingly. When the intermediate specifications can be kept small using heuristics for minimization, the state explosion problem is avoided and there are already promising experimental results for systems with an interleaving semantics, including real-time systems. This paper extends the quotienting approach to deal with a synchronous framework in the shape of state/event systems. A state/event system is a concurrent system with a set of interdependent components operating synchronously according to stimuli (input events) provided by an environment while producing output events in return for the environment. A compositional modal logic $\mathcal{M}$ suitable for expressing general safety and liveness properties subsystems is introduced. A quotient construction for bulding components from a state/event system into the specification is presented and heuristics for minimizing formulae are proposed. The techniques are demonstrated on an example. The correctness of the techniques are justified by proofs in an appendix.

[*]Basic Research in Computer Science,
Centre of the Danish National Research Foundation.

1 Introduction

Within the last decade, model checking and especially reachability checking has become a widely used technique for verifying finite state systems. However, a major problem in applying model checking on even moderate sized systems is the state explosion problem, arising from the possible combination of independent transitions. It has been shown that this problem is P-space complete, and thus in theory intractable. However, by inventing various heuristics used in analyzing and verifying systems, it has been possible to verify systems with a large number of components.

One such attack on the state explosion problem is BDD's, see [7, 9, 8, 1], which provides a canonical form for boolean formulae that is often substantially more compact than formulae on conjunctive and disjunctive normal form, and very efficient algorithms have been developed for manipulating formulae based on their BDD representation.

Another alternative is compositionality, where the motivation is to reason about the behaviour of a large system based on knowledge of its components. In those cases where a global investigation can be avoided efficiency is gained. In [4, 1, 2], compositional reasoning has proven to be a successful technique in verification of concurrent systems and embedded software. The Compositional Backwards Reachability technique (CBR) presented in [1], is used by the commercial verification tool VisualSTATE^{TM}. This tool uses the state/event model, in which a concurrent system with a set of interdependent state/event machines operate synchronously according to stimuli provided by an environment. A transition in a state/event machine is labelled with en input event e, an output event o and a guard g, which is a the set of global states in which the transition is enabled. The machines operate synchronuosly in lock–step, i.e. whenever an input e is provided by the environment all machines that are able to take a step will do so, hereby returning a set of outputs to the environment.

VisualSTATE^{TM} makes it possible to produce and verify embedded software e.g. in mobile phones. It performs reachability checks and checks for possible deadlocks. Furthermore VisualSTATE can generate code for state/event systems.

In this paper we apply the quotient framework in which the idea is to repeadetly remove one state/event machine in a parallel composition while transforming the specification accordingly. The method is applicable to verifying problems of the following form:

$$(M_1|...|M_n) \models \varphi, \tag{1}$$

where M_i is a state/event machine and φ is the property for the state/event system. A machine is removed by applying a quotient operator, $/$, to the formula, reducing the problem to

$$(M_1|...|M_{n-1}) \models \varphi/M_n, \tag{2}$$

in the sense that (1) is true if and only if (2) is.

If, after each factorization step, a set of minimization heuristics are applied on the quotient the model checking problem may be be significantly reduced.

The modal logic $\mathcal{M}'$:

$$\varphi ::= tt|ff|g|\varphi_1 \wedge \varphi_2|\varphi_1 \vee \varphi_2|\langle e\rangle\varphi|[e]\varphi|X,$$

where g is a set of global states, e is an input event, $\langle e\rangle, [e]$ denotes existential and universal quantification over events and X is an identifier, may be used to express both safety and liveness properties of state/event systems. Although the individual state/event machines have guards on their transitions, the global semantics of a state/event system will be one where all internal conditions are resolved. Another way to say this is that a state/event system as a whole is *dependency closed*.

Unfortunately, $\mathcal{M}'$ is not compositional. In the process of removing machines, the lefthand side of the verification problem above,(2), may appear not to be dependency closed. More precisely the remaining state/event machines may be contraining the machines just removed. Hence, we need an extended logic, $\mathcal{M}$ where assumptions about other components are accounted for.

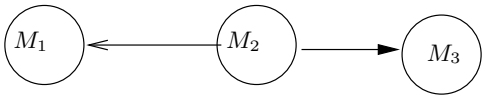

Figure 1: This figure shows the dependency graph for a system consisting of three state/event machines M_1, M_2 and M_3. An arrow from one machine M_i to another machine M_j indicates that M_i depends on M_j, i.e., that M_i has a transition with a guard that refers to a state in M_j. The system $M_1|M_2|M_3$ is dependency closed and thus the logic $\mathcal{M}'$ is ideal as a specification formalism. However, if we in the quotient procedure remove machine M_3 the remaining system $M_1|M_2$ is no longer dependency closed and the logic $\mathcal{M}$ is used instead.

To illustrate this idea we look at the subsystem $M_1|M_2|M_3$ in Figure 1. This system is dependency closed, i.e., it does not have constraints on any other machine and thus the logic $\mathcal{M}'$ is ideal for expressing properties. However, if we in the quotient procedure remove machine M_3, the remaining system $M_1|M_2$ is no longer dependency closed. More precisely, the transitions in $M_1|M_2$ may be constraining the local states of M_3. Therefore we extend the modalities in logic $\mathcal{M}'$ to the following ones: $\langle u \mapsto e\rangle$ and $[u \mapsto e]$, where u is a set of states in the context of the current subsystem. Intuitively, for a state s to satisfy a formula $\langle u \mapsto e\rangle\varphi$ it should hold that s has an e–transition which can indeed be performed when the context is in one of the states in u and such that the derived state s' satisfies φ. Similary the $xboksue\varphi$ formula is satisfied by a state s if, all e–derivatives, s', that may be reached from s under the assumption that the context is in state in u, satisfies φ.

Outline

The state/event model is introduced and formally defined in section 2. In Section 3 we introduce the logics $\mathcal{M}$ and $\mathcal{M}^*$. Section 4 presents the quotient contruction for building a state/event machine into a specification together with heuristics for minimizing formulae. An example of the use of the quotient technique is given in section 5, and section 6 draws some conclusions and depicts further work. The proofs of correctness of the quotient construction for S/E systems appear in appendix A.

2 State/event systems

This section will give an introduction to state/event systems, which will be called S/E systems in the following. A S/E system consists of n machines $M_1, ..., M_n$ over an alphabet of input events E and output events O. The presence of output events have no impact on verification at all and henceforth we will not be considering outputs in this model. Each machine M_i is a triple $(\Sigma_{\{i\}}, s_i^0, \rightarrow_{\{i\}})$ of local states, an initial state and a set of transitions. The set of transitions is a relation:

$$\rightarrow_{\{i\}} \subseteq \Sigma_{\{i\}} \times E \times G_{\{i\}} \times \Sigma_{\{i\}}$$

where $G_{\{i\}}$ is the set of guards not containing references to machine i. These guards are generated from the following simple grammar:

$$g ::= l_j = p | \neg g | g \wedge g | tt \tag{3}$$

The atomic predicate $l_j = p$ is read as "machine j is at local state p" and tt denotes a true guard. The global state set of the S/E system is the product of the local state sets:

$$\Sigma = \Sigma_{\{1\}} \times \Sigma_{\{2\}} \times ... \times \Sigma_{\{n\}}.$$

In addition to using the given syntax for guards, we find it useful to use set notation. In set notation, a guard is simply a set of allowed states. For instance, let

$$\begin{array}{rcl} \Sigma_1 & = & \{1, 2, 3\} \\ \Sigma_2 & = & \{a, b, c\} \\ \Sigma_3 & = & \{\alpha, \beta, \gamma\} \end{array} \tag{4}$$

be state spaces. Then $g = \Sigma_2 \times \{\beta\}$ would be a guard on a transition in M_1, allowing M_2 to be in any state, and requiring M_3 to be in state β. It is important to note that although a set can express more complex guards than the above syntax could, every guard in set notation must still be equivalent to some guard built from the syntax.

Projection

It is sometimes necessary to extract requirements on only a few machines from a larger guard. For instance, using the state spaces in (4), the guard $g' = \Sigma_1 \times \{b\}$ would be a condition on M_1 and M_2. With current constructs, we cannot say e.g. that M_2 in state b does not conflict with g' since formally, $b \notin g'$. We use projection to remedy this.

Definition 2.1 (Projection)
Let $g \subseteq \Sigma_I$ be a guard on machines with index set $I = \{i_1, i_2, \ldots, i_l\}$, and let $J = \{j_1, j_2, \ldots, j_m\}$ be a subset of I. Then the projection of g onto J, denoted $\Pi_J(g)$, is the guard defined by

$$\begin{aligned}\Pi_J(g) = \; & \big\{ (s_1, s_2, \cdots, s_m) \in \Sigma_J | \\ & \exists\, (t_1, t_2, \cdots, t_l) \in g : \forall p \in \{1, \cdots m\} \\ & \forall k \in \{1, \cdots, l\}\, \forall j \in J : \\ & (t_k, s_p \in \Sigma_j \;\Leftrightarrow\; t_k = s_p) \big\}\end{aligned}$$

Completeness

In our setting we only use S/E machines with complete transition relations. A transition system is complete if there is always transition that can be taken, i.e. given a state $s \in \Sigma_i$, an event $e \in E$ and the guards $g_1, ..., g_m$ on the e-transitions of s, it holds that $g_1 \vee ... \vee g_m = tt$ or, using set notation, $\bigcup_{i=\{1..m\}} g_i = \Sigma_{\{1...n\}\setminus\{i\}}$. An incomplete transistion relation can be made complete in the following way: Suppose a state s is not complete. Then by adding the transition $s \xrightarrow{e, \neg g_1 \wedge ... \wedge \neg g_m} s$ we obtain a system having exactly the same set of reachable states.

Composition

A S/E system can be composed of individual S/E machines.

Definition 2.2 (Composition)
Let M_I and M_J be S/E machines, where $I, J \subseteq \{1 \ldots n\}$ and $I \cap J = \emptyset$ Then we define the composition $M_{I \cup J}$ to be $(\Sigma_I \times \Sigma_J, s_I^0 \times s_J^0, \rightarrow_{I \cup J})$ where $\rightarrow_{I \cup J}$ is defined as follows:

$$\frac{\bar{s} \xrightarrow{e\ g}_I \bar{s}' \quad \bar{t} \xrightarrow{e\ h}_J \bar{t}'}{\bar{s}\bar{t} \xrightarrow{e\ \Pi_{\overline{I \cup J}}(\{\bar{s}\} \times g \cap h \times \{\bar{t}\})}_{I \cup J} \bar{s}'\bar{t}'}$$

$$\text{where } \bar{s}, \bar{s}' \in \Sigma_I \text{ and } \bar{t}, \bar{t}' \in \Sigma_J$$

It can easily be shown that completeness is preserved by composition.

Definition 2.3 (Dependency closed system)
Let $M = (\Sigma_I, s_o, \rightarrow_I)$ be a S/E machine. Then, M is dependency closed if all guards in $\rightarrow_I$ are true.

3 Logic

In order to express interesting properties of S/E-systems, we employ a series of modal logics $\mathcal{M}$, in which there is a specific logic $\mathcal{M}_I$ for each set I of machine indices. Each specific logic applies to machines with the same index set as the logic. This allows us to apply the quotient operator to a property, that a set of machines must satisfy in order to obtain a property that a subset of those machines must satisfy.

Syntax

Let I be a set of machine indices. Then we define logic $\mathcal{M}_I$ as

$$\varphi ::= tt|ff|g|\varphi_1 \wedge \varphi_2|\varphi_1 \vee \varphi_2|\langle u \mapsto e\rangle\varphi|[u \mapsto e]\varphi|X,$$

where X is an identifier, e is an event, $g \subseteq \Sigma_I$ and $u \subseteq \Sigma_{\bar{I}}$

Semantics

Let I denote a set of machine indices, and φ be a formula in logic $\mathcal{M}_I$ and let $\bar{s}$ be a state in M_I. Then the statement $\bar{s} \models_I \varphi$ is read "state $\bar{s}$ satisfies formula φ under logic $\mathcal{M}_I$. Usually the specific logic will be clear from the context, so we simply write $\bar{s} \models \varphi$. The semantics for $\bar{s} \models \varphi$ is given by

$$\begin{array}{lll}
\bar{s} \models & tt & \\
\bar{s} \models & g & \text{iff} \quad \bar{s} \in g \\
\bar{s} \models & \varphi_1 \wedge \varphi_2 & \text{iff} \quad \bar{s} \models \varphi_1 \text{ and } \bar{s} \models \varphi_2 \\
\bar{s} \models & \varphi_1 \vee \varphi_2 & \text{iff} \quad \bar{s} \models \varphi_1 \text{ or } \bar{s} \models \varphi_2 \\
\bar{s} \models & \langle u \mapsto e\rangle\varphi & \text{iff} \quad \exists \bar{s}', g : \bar{s} \xrightarrow{e\ g} \bar{s}' \text{ s.t. } u \subseteq g \wedge \bar{s}' \models \varphi \\
\bar{s} \models & [u \mapsto e]\varphi & \text{iff} \quad \forall \bar{s}', g : (\bar{s} \xrightarrow{e\ g} \bar{s}' \text{ s.t. } u \subseteq g) \Rightarrow \bar{s}' \models \varphi \\
\bar{s} \models & X & \text{iff} \quad \bar{s} \models \mathcal{D}(X)
\end{array}$$

We will use the statement $M_I \models \varphi$ to mean $\bar{s}_I^0 \models \varphi$, where $\bar{s}_I^0$ is the initial state of M_I.

In our logic we express reachability properties as follows: Let $g \in G$. Then the property that a state satisfying g is reachable is expressed as the minimal fixpoint solution to the following equation:

$$Reach(g) = g \vee \bigvee_{e \in E} \langle e\rangle Reach(g)$$

4 Quotienting

This section will describe the quotient technique, and the use of equation systems and minimization heuristics when utilizing the technique.

Defining the quotient operator presents two challenges:

1. The machine being factored out may have guards restricting other machines.
2. There may be other machines that guard the machine being factored out (we say that the machine is *guarded*).

In the following, we formally define and state the correctness of the quotient operator.

Formal definition of quotient

Definition 4.1 (Quotient operator)
Let I and J be sets of machine indices where $J \subseteq I$. Let $\bar{s} \in \Sigma_J$, and let φ be a formula in $\mathcal{M}_I$. Then, $\varphi/\bar{s}$ is a formula in $\mathcal{M}_{I\setminus J}$. It is defined inductively on the structure of φ as follows:

$$
\begin{aligned}
g/\bar{s} &= \{\bar{t} \in \Sigma_{I\setminus J} | \{\bar{t}\} \times \{\bar{s}\} \subseteq g\} \\
(\varphi_1 \wedge \varphi_2)/\bar{s} &= \varphi_1/\bar{s} \wedge \varphi_2/\bar{s} \\
(\varphi_1 \vee \varphi_2)/\bar{s} &= \varphi_1/\bar{s} \vee \varphi_2/\bar{s} \\
(\langle u \mapsto e \rangle \varphi)/\bar{s} &= \bigvee_{j | \bar{s} \xrightarrow{e\ g_j}_J \bar{s}_j} \Big(\mu_j \wedge \langle u \times \bar{s} \mapsto e \rangle (\varphi/\bar{s}_j) \Big) \\
([u \mapsto e]\varphi)/\bar{s} &= \bigwedge_{j | \bar{s} \xrightarrow{e\ g_j}_J \bar{s}_j} \Big(\mu_j \Rightarrow [u \times \bar{s} \mapsto e] (\varphi/\bar{s}_j) \Big) \\
X/\bar{s} &= X^{\bar{s}}, \text{ where } X^{\bar{s}} = \mathcal{D}(X)/\bar{s}
\end{aligned}
$$

where $\mu_j = \Pi_{I\setminus J}\Big([u \times \Sigma_{I\setminus J}] \cap g_j \Big)$

4.1 Equation systems

When utilizing the quotient technique, it is necessary to use equation systems. When wanting to factor out some machine $M = \big\langle (s_0, ..., s_k), s_0, \rightarrow \big\rangle$, it is necessary to factor out each state in the S/E machine M. This is done with equation systems. Let X be an identifier and $\varphi \in \mathcal{L}$

then

$$X/M = \begin{cases} x_0 & = & \varphi/s_0 \\ x_1 & = & \varphi/s_1 \\ \vdots & = & \vdots \\ x_k & = & \varphi/s_k \end{cases}$$

where x_0 is the "top formula" which represents the result we want to compute. We might obtain one or more equations that evaluate to tt or ff.

4.2 Minimization heuristics

In the following let $\varphi, g \in \mathcal{M}_I, u \subseteq \Sigma_{\bar{I}}$ and $e \in E$.

Simple evaluation: $\varphi \vee tt \mapsto tt$ and $\varphi \wedge tt \mapsto \varphi$

Trivial diamond elimination: $\langle u \mapsto e \rangle tt \mapsto tt$

Trivial box elimination: $[u \mapsto e] ff \mapsto ff$

Trivial diamond elimination and trivial box elimination are possible because we use complete S/E systems. According to the definition of completeness a complete S/E system is able to perform a transition at any time.

Dead Event Elimination: $\langle u \mapsto e \rangle \varphi \mapsto \varphi$ if $\bar{s} \xrightarrow{e\ g}_I \bar{s}$ for all $\bar{s} \in \Sigma_I$.

Dead Event Elimination may be a powerful reduction heuristic in those cases where we have sort–information of the system in focus.

Trivial Disjunction Elimination: $\bigvee_i g_i \mapsto tt$ if $\bigvee_i g_i \equiv tt$

Definition 4.2 (Context Equivalence)
Let G be the set of guards in machines not yet factored out, and let u_1 and u_2 be sets of states in the subsystem already factored out. Then u_1 and u_2 are said to be context equivalent modulo G, $u_1 =_G u_2$, if the following holds:

$$\forall g \in G : u_1 \subseteq g \text{ iff } u_2 \subseteq g$$

Context Dependent Reduction I: Let G be the set of guards in machines $M_i, i \in \bar{I}$. Then

$$\langle u_1 \mapsto e \rangle \varphi \vee \langle u_2 \mapsto e \rangle \varphi \mapsto \langle u_1 \mapsto e \rangle \varphi \text{ if } u_1 =_G u_2.$$

Context Dependent Reduction II: Let G be the set of guards in machines $M_i, i \in \bar{I}$. Then

$$[u_1 \mapsto e] \varphi \vee [u_2 \mapsto e] \varphi \mapsto [u_1 \mapsto e] \varphi \text{ if } u_1 =_G u_2.$$

Recursion Elimination: Let X be an identifier. Then, $X = \langle e \rangle X \mapsto ff$ when computing the minimal fixpoint.

Theorem 4.3 (Correctness of quotient)
Let $M_1|\cdots|M_n$ be a S/E system, let $I = \{1...i\}$ be a set of machine indices, and let $\bar{s} \in \Sigma_{I\setminus\{i\}}$, $s_i \in \Sigma_i$ and $\varphi \in \mathcal{M}_I$. Then it holds that

$$(\bar{s}, s_i) \models \varphi \Longleftrightarrow \bar{s} \models \underbrace{\varphi/s_i}_{\mathcal{L}_{I\setminus\{i\}}}$$

5 Example

As an example we shalle consider a lecture room with n blackboards $B_1, \ldots, B_n$ which are placed side by side and are able to move up and down independently, see Figure 2. The

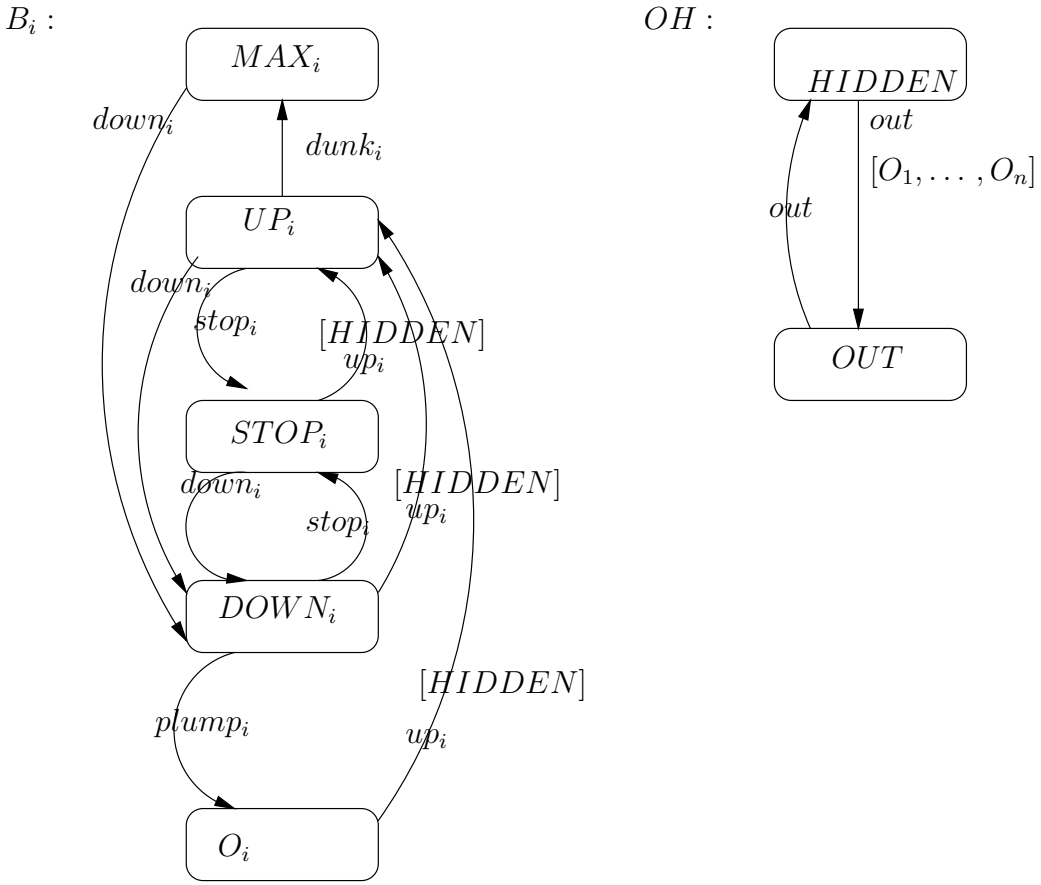

Figure 2: The i'th board B_i and the overhead projector OH.

$i'th$ board has five states: MAX_i for being in the higehst possible position, UP_i represents upwards movement, $STOP_i$ for not moving at all (also $STOP_i$ is the initial state), $DOWN_i$

for moving downwards and finally O_i for being in the lowest possible position. On the wall behind the boards there is a wide overhead screen (OH) which is initially $HIDDEN$, i.e. its alignement is vertical but may "tip" out to a suitable angle when all the boards are in position O_i. Similarly, the boards can only move up when the overhead screen is $HIDDEN$.

We want to prove that it is possible to get the overhead screen to the position OUT. We therefore get the following specification:

$$X = [OH@OUT] \vee \bigvee_{e \in \mathcal{E} \cup out} \langle e \rangle X$$

where $E_i = \{up_i, down_i, stop_i, plump_i, dunk_i\}$ and $\mathcal{E} = \bigcup_{i=1...n} E_i$.

Now, the idea is to first factor out the overhead projector followed by all the boards one by one. It turns out that it is possible to keep the quotient down to only one single equation after eacf factorization step, and thus making the verification very feasible.

Factoring out the overhead projector gives two variables X^H and X^O:

$$\begin{aligned} X^H &= [\bigwedge_{i=1...n} B_i@O_i \wedge \langle H \mapsto out \rangle X_O] \vee \bigvee_{e \neq out} \langle H \mapsto e \rangle X_H \\ X^O &= (OH@OUT)/OUT = tt \end{aligned}$$

By substituting tt for X^O in X^H and applying trivial diamond elimination we get a new specification, X, for the boards

$$X = \bigwedge_{i=1...n} B_i@O_i \vee \bigvee_{e \neq out} \langle H \mapsto e \rangle X$$

Here H denotes $\{HIDDEN\}$

Now, let us factor out the $i'th$ board B_i. We get the following five equations:

$$\begin{aligned} X^H/MAX_i &= ff \vee \langle (MAX_i, H) \mapsto down_i \rangle (X^H/DOWN_i) \vee \\ &\quad \bigvee_{e_i \in E_i \setminus \{down_i\}} \langle (MAX_i, H) \mapsto e_i \rangle (X^H/MAX_i) \vee \\ &\quad \bigvee_{e_j \in \mathcal{E} \setminus E_i} \langle (MAX_i, H) \mapsto e_j \rangle (X^H/MAX_i) \end{aligned}$$

$$\begin{aligned} X^H/UP_i &= ff \vee \langle (UP_i, H) \mapsto down_i \rangle (X^H/DOWN_i) \vee \\ &\quad \langle (UP_i, H) \mapsto dunk_i \rangle (X^H/MAX_i) \vee \\ &\quad \langle (UP_i, H) \mapsto stop_i \rangle (X^H/STOP_i) \vee \\ &\quad \bigvee_{e_i \in E_i \setminus \{down_i, dunk_i, stop_i\}} \langle (UP_i, H) \mapsto e_i \rangle (X^H/UP_i) \vee \\ &\quad \bigvee_{e_j \in \mathcal{E} \setminus E_i} \langle (UP_i, H) \mapsto e_j \rangle (X^H/UP_i) \end{aligned}$$

$$
\begin{aligned}
X^H/STOP_i \;=\;& ff \vee \langle (STOP_i, H) \mapsto up_i \rangle (X^H/UP_i) \vee \\
& \langle (STOP_i, H) \mapsto down_i \rangle (X^H/DOWN_i) \vee \\
& \bigvee_{e_i \in E_i \setminus \{up_i, down_i\}} \langle (STOP_i, H) \mapsto e_i \rangle (X^H/STOP_i) \vee \\
& \bigvee_{e_j \in \mathcal{E} \setminus E_i} \langle (STOP_i, H) \mapsto e_j \rangle (X^H/STOP_i)
\end{aligned}
$$

$$
\begin{aligned}
X^H/DOWN_i \;=\;& ff \vee \langle (DOWN_i, H) \mapsto up_i \rangle (X^H/UP_i) \vee \\
& \langle (DOWN_i, H) \mapsto stop_i \rangle (X^H/STOP_i) \vee \\
& \langle (DOWN_i, H) \mapsto plump_i \rangle (X^H/O_i) \vee \\
& \bigvee_{e_i \in E_i \setminus \{up_i, stop_i, plump_i\}} \langle (DOWN_i, H) \mapsto e_i \rangle (X^H/DOWN_i) \vee \\
& \bigvee_{e_j \in \mathcal{E} \setminus E_i} \langle (DOWN_i, H) \mapsto e_j \rangle (X^H/DOWN_i)
\end{aligned}
$$

$$
\begin{aligned}
X^H/O_i \;=\;& ff \vee \langle (O_i, H) \mapsto up_i \rangle (X^H/UP_i) \vee \\
& \bigvee_{e_i \in E_i \setminus \{up_i\}} \langle (MAX_i, H) \mapsto e_i \rangle (X^H/O_i) \vee \\
& \bigvee_{e_j \in \mathcal{E} \setminus E_i} \langle (O_i, H) \mapsto e_j \rangle (X^H/O_i) \vee \\
& \bigwedge_{j \in \{1 \ldots n\} \setminus \{i\}} B_j @ O_j
\end{aligned}
$$

Now, since no board $B_j, j \neq i$ will ever use any of the events in E_i we may use the principle of Dead Event Elimination to simplify the equationa above. Moreover, we observe that all lefthand sides also appear unguarded on most righthand sides and thus we may conclude that:

$$
X^H/MAX_i \Leftrightarrow X^H/UP_i \Leftrightarrow X^H/STOP_i \Leftrightarrow X^H/DOWN_i \Leftrightarrow X^H/O_i
$$

Hence, we get the following new specification:

$$
X = \bigwedge_{j \in \{1 \ldots n\} \setminus \{i\}} B_j @ O_j \vee \bigvee_{e_j \in \mathcal{E} \setminus E_i} \langle (L_i, H) \mapsto e_j \rangle X
$$

where $L_i \in \{MAX_i, UP_i, STOP_i, DOWN_i, O_i\}$.

No board B_j ever constrains board B_i, i.e. B_j does not distinguish between the states MAX_i, UP_i, $STOP_i$, $DOWN_i$, and O_i and thus we may reduce X to

$$X = \bigwedge_{j \in \{1 \ldots n\} \setminus \{i\}} B_j @ O_j \vee \bigvee_{e_j \in \mathcal{E} \setminus E_i} \langle H \mapsto e_j \rangle X$$

Here H denotes $E_i \times \{HIDDEN\}$.

Thus, after having factored out all the boards we will obtain the single equation

$$Y = \bigwedge_{j \in \emptyset} B_j @ O_j \vee \bigvee_{e_j \in \emptyset} \langle H \mapsto e_j \rangle Y \mapsto tt \vee ff \mapsto tt$$

Thus we have succeded in verifying the system while avoiding the burden of the state explosion problem.

6 Conclusion and further work

In this paper we have addressed the state explosion problem by defining and proving the correctness of the quotient technique in Left Restricting state/event systems. We have found the simple diamond operator $\langle e \rangle$ insufficient to deal with systems that are not dependency closed. Therefore we have developed an extended modal logic featuring an extended diamond operator $\langle \bar{u} \mapsto e \rangle$. Our work should provide a good framework for extending the quotient technique to deal with cyclic dependencies.

Hierachical systems is a new feature in VisualSTATE where states in the system can be expressed as subsystems. Currently the compositional backwards reachability approach used in VisualSTATE fails to handle these systems effectively. The quotient technique might be a solution to this problem.

So far implementation of the technique and the gathering of experimental results is subject to further work.

References

[1] Lind-Nielsen, J., Andersen, H.R., Behrmann, G., Hulgaard, H., Kristoffersen, K.J., Larsen, K.G.
Verfication of Large State/Event Systems Using Compositionality and Dependency Analysis
Presented at TACAS'98 and published in LNCS 1384, Springer Verlag 1998

[2] Behrmann, G., Kristoffersen, K.J, Larsen, K.G.
Context Dependent Minimization of State/Event Systems
Presented at NWPT'97 and appears in Proceedings of the Estonian Academy of Sciences.

[3] Kåre Jelling Kristoffersen
Compositional Verfication of Concurrent Systems
PhD thesis, Aalborg University august 1998
ISSN 1397-8640

[4] Henrik Reif Andersen
Partial Model Checking (Extended Abstract)
Appears in proceedings of LICS95 (pp.398-407), IEEE Computer Society Press

[5] Kim Guldstrand Larsen
From Modal Logic to Process Algebra
Aalborg University Center, September 1987

[6] Hans Hüttel, Kim Guldstrand Larsen
Bevissystemer for opfyldelighed i Hennessy-Milner-logik med rekursion
Aalborg University Center, 18. Januar 1999

[7] R. E. Bryant. Graph–based algorithms for boolean function manipulation. *IEEE Transactions on Computers*, C–35(8):677–691, 1986.

[8] J. R. Burch, E. M. Clarke, K. L. McMillan, D. L. Dill, and L. J. Hwang. Symbolic model checking: 10^{20} states and beyond. *Information and Computation*, 98(2):142–170, June 1992.

[9] K. L. McMillan. Symbolic model checking: An approach to the State Explosion Problem. *Kluwer Academic Publishers*, 1993.

A Proving correctness of the quotient technique

To ensure that the the quotient technique yields correct results, we need to prove that if we have a compound state $\bar{s} = (s_1, \ldots, s_i, \ldots, s_n)$ in some set of machines, it holds that $\bar{s}$ satisfies a formula φ if and only if the smaller state $(s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_n)$ satisfies $\frac{\varphi}{s_i}$. However, we look at the simpler case where the S/E system is Left Restricting.

A.1 Factoring out unguarded machines

Theorem A.1 (Correctness of quotient)
Let $M_1|\cdots|M_n$ be a Left Restricting S/E system, let $I = \{1...i\}$ be a set of machine indices, and let $\bar{s} \in \Sigma_{I\setminus\{i\}}$, $s_i \in \Sigma_i$ and $\varphi \in \mathcal{L}_I$. Then it holds that

$$(\bar{s}, s_i) \models \varphi \iff \bar{s} \models \underbrace{\varphi/s_i}_{\mathcal{L}_{I\setminus\{i\}}}$$

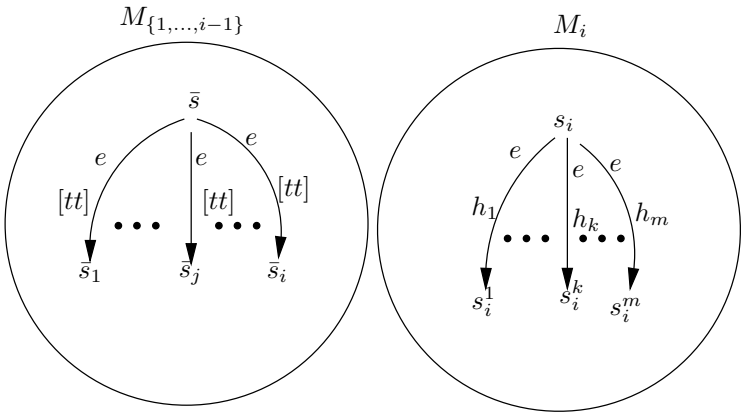

Figure 3: Illustrates the S/E system used in the proof of correctness of quotient. Here we focus on e-transitions from $\bar{s}$ and s_i. The guards on e-transitions from $\bar{s}$ are tt since this is a Left Restricting S/E system, and $M_{\{1,\ldots,i-1\}}$ is the leftmost S/E machine. The guards on e-transitions from s_i need not be tt.

Proof
This proof is by structural induction in φ. For convenience, the states under consideration are illustrated in figure 3.

Induction hypothesis: Let $M_1|\cdots|M_n$ be a Left Restricting S/E system. For any index $i \in \{1, \ldots, n\}$, let $\bar{t}$ and t_i be states in $M_{\{1,\ldots,i-1\}}$ and M_i, respectively. Then,

$$(\bar{t}, t_i) \models \varphi_1 \quad \Leftrightarrow \quad \bar{t} \models \tfrac{\varphi_1}{t_i}$$
$$(\bar{t}, t_i) \models \varphi_2 \quad \Leftrightarrow \quad \bar{t} \models \tfrac{\varphi_2}{t_i}$$

Base step: $\varphi = g$
Show $(\bar{s}, s_i) \models g \Leftrightarrow \bar{s} \models \frac{g}{s_i}$.
We show the biimplication by starting from the right:

$$\begin{array}{rl} & \bar{s} \models \frac{g}{s_i} \\ \Leftrightarrow & \bar{s} \models \left\{\bar{t} | \bar{t} \times \{s_i\} \in g\right\} \\ \Leftrightarrow & \bar{s} \in \left\{\bar{t} | \bar{t} \times \{s_i\} \in g\right\} \\ \Leftrightarrow & (\bar{s}, s_i) \in g \\ \Leftrightarrow & (\bar{s}, s_i) \models g \end{array}$$

Inductive step: $\varphi = g \wedge \varphi_1$
Show $(\bar{s}, s_i) \models g \wedge \varphi_1 \Leftrightarrow \bar{s} \models \frac{g \wedge \varphi_1}{s_i}$.
We show the biimplication by starting from the right. IH denotes application of the induction hypothesis.

$$\begin{array}{rl} & \bar{s} \models \frac{g \wedge \varphi_1}{s_i} \\ \Leftrightarrow & \bar{s} \models \frac{g}{s_i} \wedge \bar{s} \models \frac{\varphi_1}{s_i} \\ \overset{\text{IH}}{\Leftrightarrow} & (\bar{s}, s_i) \models g \wedge (\bar{s}, s_i) \models \varphi_1 \\ \Leftrightarrow & (\bar{s}, s_i) \models g \wedge \varphi_1 \end{array}$$

Inductive step: $\varphi = \varphi_1 \vee \varphi_2$
Show $(\bar{s}, s_i) \models \varphi_1 \vee \varphi_2 \Leftrightarrow \bar{s} \models \frac{\varphi_1 \vee \varphi_2}{s_i}$.

$$\begin{array}{rl} & \bar{s} \models \frac{\varphi_1 \vee \varphi_2}{s_i} \\ \Leftrightarrow & \bar{s} \models \frac{\varphi_1}{s_i} \vee \bar{s} \models \frac{\varphi_2}{s_i} \\ \overset{\text{IH}}{\Leftrightarrow} & (\bar{s}, s_i) \models \varphi_1 \vee (\bar{s}, s_i) \models \varphi_2 \\ \Leftrightarrow & (\bar{s}, s_i) \models \varphi_1 \vee \varphi_2 \end{array}$$

Inductive step: $\varphi = \langle e \rangle \varphi_1$
Show $(\bar{s}, s_i) \models \langle e \rangle \varphi_1 \Leftrightarrow \bar{s} \models \frac{\langle e \rangle \varphi_1}{s_i}$.
We show the biimplication from left to right. From the semantics of diamond and the fact that the system is Left Restricting we have,

$$\begin{array}{rl} & (\bar{s}, s_i) \models \langle e \rangle \varphi_1 \\ \Leftrightarrow & \exists j, k : \left[(\bar{s}, s_i) \xrightarrow{e\ [tt]} \left(\bar{s}_j, s_i^k\right) \right. \\ & \left. \wedge \left(\bar{s}_j, s_i^k\right) \models \varphi_1 \right] \\ \overset{\text{IH}}{\Leftrightarrow} & \exists j, k : \left[\underline{\bar{s} \xrightarrow{e\ [tt]} \bar{s}_j} \wedge s_i \xrightarrow{e\ h_k} s_i^k \wedge \bar{s} \in h_k \right. \\ & \left. \wedge \bar{s}_j \models \underline{\frac{\varphi_1}{s_i^k}} \right] \end{array}$$

By combining the underlined statements to a diamond formula, we continue the biimplication:

$$\begin{array}{rl} & \exists k : \left[\bar{s} \models \left(h_k \wedge \langle e \rangle \frac{\varphi_1}{s_i^k} \right) \wedge s_i \xrightarrow{e\ h_k} s_i^k \right] \\ \Leftrightarrow & \bar{s} \models \bigvee_{k | s_i \xrightarrow{e\ h_k} s_i^k} h_k \wedge \langle e \rangle \frac{\varphi_1}{s_i} \\ \Leftrightarrow & (\bar{s}) \models \frac{\langle e \rangle \varphi_1}{s_i} \end{array}$$

Which concludes the proof. □ **Proof**

Again, structural induction on φ is used. Since the proof is largely similar to that of Theorem A.1, only the case of extended diamond is shown. For convenience, the states under consideration are illustrated in figure 4.

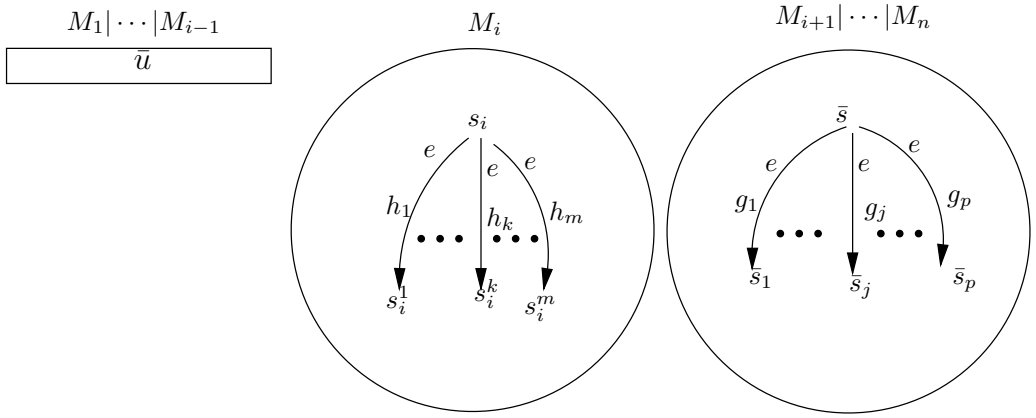

Figure 4: Illustrates the situation where machines M_1 to M_{i-1} have been factored out. Only the e-transitions from state s_i in M_i and state $\bar{s}$ in $M_{i+1}|\cdots|M_n$ are shown. The rectangle represents the information $\bar{u}$ provides about the state of $M_1|\cdots|M_{i-1}$.

Induction Hypothesis: Let $M_1|\cdots|M_n$ be a Left Restricting S/E system. For any index $i \in \{1,\ldots,n\}$, let $\bar{t}_i$ and $\bar{t}$ be states in M_i and $M_{\{i,\ldots,n\}}$ respectively. Then,

$$(t_i, \bar{t}) \models \varphi_1 \quad \Leftrightarrow \quad \bar{t} \models \tfrac{\varphi_1}{t_i}$$

Case $\varphi = \langle \bar{u} \mapsto e \rangle \varphi_1$
Show $(s_i, \bar{s}) \models \langle \bar{u} \mapsto e \rangle \varphi_1 \Longleftrightarrow \bar{s} \models \frac{\langle \bar{u} \mapsto e \rangle \varphi_1}{\bar{s}_i}$.

We show the biimplication by beginning with the left-hand side: From the semantics of extended diamond, we have:

$$\begin{array}{ll} & (s_i, \bar{s}) \models \langle \bar{u} \mapsto e \rangle \varphi_1 \\ \Leftrightarrow & \exists k, j : \big[(s_i, \bar{s}) \xrightarrow{e\, f} \left(s_i^k, \bar{s}_j\right) \\ & \wedge \left(s_i^k, \bar{s}_j\right) \models \varphi_1 \wedge \bar{u} \in f \big] \end{array}$$

where $f = \Pi_{\bar{I}}\Big(\left(\Pi_{\bar{I}}\left(h_k\right) \times \{s_i\}\right) \cap g_j \Big)$ is the guard resulting from composing $M_{\{i\}}$ and $M_{I\setminus\{i\}}$ according to Definition 2.2. Note that $\bar{u}$ lies in f if and only if $\bar{u} \in h_k$ and $\bar{u} \in \Pi_{\bar{I}}(g_j)$ and

$s_i \in \Pi_{\{i\}}(g_j)$. Using also the inductive hypothesis, this expands the biimplication to include

$$\begin{array}{l}\exists k, j : \big[s_i \xrightarrow{e\ h_k} s_i^k \wedge \ \underline{\bar{s} \xrightarrow{e\ g_j} \bar{s}_j} \\ \wedge \underline{\Big(\bar{s}_j \models \frac{\varphi_1}{s_i^k}\Big)} \wedge \bar{u} \in h_k \\ \wedge \underline{\bar{u} \in \Pi_{\bar{I}}(g_j) \wedge s_i \in \Pi_{\{i\}}(g_j)}\big]\end{array}$$

Observing that $\big(\bar{u} \in \Pi_{\bar{I}}(g_j) \wedge s_i \in \Pi_{\{i\}}(g_j)\big) \ \Leftrightarrow \ \{\bar{u}\} \times \{s_i\} \in g_j$, we combine the underlined statements:

$$\begin{array}{l}\exists k : \big[s_i \xrightarrow{e\ h_k} s_i^k \wedge \bar{u} \in h_k \\ \wedge \bar{s} \models \langle \{\bar{u}\} \times \{s_i\} \mapsto e \rangle \frac{\varphi_1}{s_i^k}\big]\end{array}$$

Let $\mu_k = \begin{cases} \Sigma_{I \setminus \{i\}} & , \ \bar{u} \in h_k \\ \emptyset & , \ \text{else} \end{cases}$.

We now use the fact that $\bar{s} \models \mu_k$ if and only if $\bar{u} \in h_k$, and at the same time change the existential quantifier into a disjunction. The remaining steps are as follows:

$$\begin{aligned} & \bigvee_{k | s_i \xrightarrow{e\ h_k} s_i^k} \bar{s} \models \big(\mu_k \wedge \langle \bar{u} \times \{s_i\} \mapsto e \rangle \tfrac{\varphi_1}{s_i^k}\big) \\ \Leftrightarrow\ & \bar{s} \models \bigvee_{k | s_i \xrightarrow{e\ h_k} s_i^k} \big(\mu_k \wedge \langle \bar{u} \times \{s_i\} \mapsto e \rangle \tfrac{\varphi_1}{s_i^k}\big) \\ \Leftrightarrow\ & \bar{s} \models \tfrac{\langle \bar{u} \mapsto e \rangle \varphi_1}{s_i} \end{aligned}$$

□

Recent BRICS Report Series Publications

RS-99-41 **Nicky O. Bodentien, Jacob Vestergaard, Jakob Friis, Kåre J. Kristoffersen, and Kim G. Larsen.** ***Verification of State/Event Systems by Quotienting*****. December 1999. 17 pp. Presented at** ***Nordic Workshop in Programming Theory*****, Uppsala, Sweden, October 6–8, 1999.**

RS-99-40 **Bernd Grobauer and Zhe Yang.** ***The Second Futamura Projection for Type-Directed Partial Evaluation*****. November 1999. Extended version of an article to appear in Lawall, editor,** ***ACM SIGPLAN Workshop on Partial Evaluation and Semantics-Based Program Manipulation*****, PEPM '00 Proceedings, 2000.**

RS-99-39 **Romeo Rizzi.** ***On the Steiner Tree*** $\frac{3}{2}$***-Approximation for Quasi-Bipartite Graphs*****. November 1999. 6 pp.**

RS-99-38 **Romeo Rizzi.** ***Linear Time Recognition of*** P_4***-Indifferent Graphs*****. November 1999. 11 pp.**

RS-99-37 **Tibor Jordán.** ***Constrained Edge-Splitting Problems*****. November 1999. 23 pp. A preliminary version with the title** ***Edge-Splitting Problems with Demands*** **appeared in Cornujols, Burkard and Wöginger, editors,** ***Integer Programming and Combinatorial Optimization: 7th International Conference*****, IPCO '99 Proceedings, LNCS 1610, 1999, pages 273–288.**

RS-99-36 **Gian Luca Cattani and Glynn Winskel.** ***Presheaf Models for CCS-like Languages*****. November 1999. ii+46 pp.**

RS-99-35 **Tibor Jordán and Zoltán Szigeti.** ***Detachments Preserving Local Edge-Connectivity of Graphs*****. November 1999. 16 pp.**

RS-99-34 **Flemming Friche Rodler.** ***Wavelet Based 3D Compression for Very Large Volume Data Supporting Fast Random Access.*** **October 1999. 36 pp.**

RS-99-33 **Luca Aceto, Zoltán Ésik, and Anna Ingólfsdóttir.** ***The Max-Plus Algebra of the Natural Numbers has no Finite Equational Basis*****. October 1999. 25 pp. To appear in** ***Theoretical Computer Science*****.**