Model Reduction of Modules for State-Event Temporal Logics

M. Lawford[1], *J.S. Ostroff*[2] *and W.M. Wonham*[1]
[1]*Dept. of Elec. & Comp. Eng., Univ. of Toronto, Toronto, ON M5S 1A4*
[2]*Dept. of Computer Science, York Univ., North York, ON, M3J 1P3*

Abstract

In many Discrete-Event Systems (DES) both state and event information are of importance to the systems designer. Logics such as Ostroff's RTTL allow for the specification and verification of a system's state-event behavior. To make realistic problems amenable to analysis, a designer must typically decompose the system into subsystems (modules) and use algebraic abstraction (quotient systems) to obtain hierarchical system models that preserve the properties to be verified. In this paper we use state-event observational equivalence to perform model reduction for a subclass of formulas of state-event linear temporal logics, with particular attention being paid to a discrete time temporal logic that is a simplification of RTTL. The reduction technique allows limited use of immediate operators.

Keywords

Event structures, temporal logic, real-time

1 INTRODUCTION

In this paper we utilize algebraic state-event structures to model systems together with state-event temporal logics as a means of specification. The main contribution of the paper is a compositionally consistent model reduction technique for a class of "state-event stuttering invariant" temporal formulas. In particular, the method provides a means of "weak" model reduction for a discrete time temporal logic that is a simplification of Ostroff's RTTL [17]. We begin by justifying our choices of state-event and discrete time settings before outlining the sense in which our model reduction technique is both weak and compositionally consistent.

While it is possible to represent systems using only state information or only event information, there are many applications where the use of both state and event information is quite natural. In our state-event setting, the use of labeled transition relations permits the application of synchronous composition operators, thereby allowing interacting modules to perform synchronous execution of shared events such as *tick*s of a global discrete clock [17]. The *tick* events provide concurrent systems with a uniform notion of time, without the restrictiveness of the clock driven models such as [8] where one transition is one time step. An example use of *tick* events is Ostroff's RG2 graphs [18] that are employed for model checking Real-Time

Temporal Logic (RTTL) properties. RG2 graphs use event information to reduce infinite state timed systems to finite state systems while preserving the relative timing of state changes and event outputs through the use of "*tick*" transitions. Discrete time models are sufficiently accurate in many instances, particularly when dealing with digital control systems that sample their inputs (eg. [13]). In [13] the authors argue that discrete time models such as Ostroff's Timed Transition Models (TTMs) [17] allow for a straight forward application of well known process algebraic equivalences such as observation (bisimulation) equivalence from Milner's CCS [16]. On the other hand continuous time extensions of CCS such as [21] lack the abstracting power of a congruence relation like weak observation congruence [16] due to technical difficulties associated with their continuous time semantics. The addition of event information is also crucial for performing synchronous composition of systems and thereby performing supervisory control through the disablement of controllable events opening up the possibility of exploiting the synthesis techniques of the supervisory control community [20] to meet temporal specifications.

While symbolic model checking techniques have proven effective for some very large systems [5], these systems typically come from the digital hardware domain and have a great deal of regularity in their state transition structure that can be exploited by the symbolic techniques to obtain compact representations of large systems. To model check large systems lacking in symmetry or larger digital hardware systems, one must also perform some sort of model reduction.

In model reduction to facilitate the verification process, or even make the problem tractable, a reduced model is obtained such that if the reduced model satisfies the temporal formulas under investigation then the original system satisfies the temporal formulas. If the model checking of a formula on the reduced model provides a definitive answer regarding the satisfaction of the formula in the original system, we say the reduction technique is *exact.* If this model reduction technique is performed so that the mutual satisfaction of formulas is only guaranteed for an explicit finite set of formulas, we say that the method is a *formula specific* model reduction technique. But if, as in this paper, the method always guarantees the mutual satisfaction of a class of temporal formulas, we refer to the technique as being *formula independent.* In addition to preserving the truth values of a particular class of temporal formulas, the model reduction technique presented here is "compositionally consistent" in the sense that for any formula from a defined class of formulas, the composition of two reduced models satisfies the formula iff the composition of the two original systems satisfies the formula.

Our comparison of previous works with the work at hand will also make distinctions between "strong" and "weak" model reduction techniques. In a strong reduction technique, a single transition in the original system model results in a single transition in the reduced model. In weak model reduction techniques, a single transition may be used by the reduced model to represent a finite sequence of transitions in the original system model. The result is that weak model reduction techniques tend to achieve a greater reduction in state size at the expense of preserving the truth values of fewer formulas and requiring greater computational effort to compute the reduced system. In concurrent systems built from interacting modules, we are interested in specifying a module's observable behavior or "interface" with other systems. If two modules produce identical behavior at their interfaces and differ only in their internal behavior, then they should satisfy the same interface specification. While many temporal logics have been successfully used to specify systems' behaviors, straight forward application of temporal logics is often too discriminating with respect to the internal actions of concurrent systems. Since we want to reason about observed events and changes in the system's state output, we define a class of state-event stuttering invariant formulas which is similar to the stuttering invariant formulas of [15] with some key differences as a result of our state-event setting.

Methods based upon abstract interpretations such as [2], [7], provide examples of strong, formula specific model reduction. Although they are "strong" techniques, these methods can provide a significant reduction in state size by an appropriate choice of abstraction. The development of the abstract model can be an iterative process, with the mapping between concrete and abstract domains being refined when there is insufficient information at the abstract level to determine the truth value of one of the formulas of interest. The creation of these abstractions typically requires some insight from the systems designer.

All of the above formula dependent techniques suffer from an inability to guarantee compositional consistency. Hence, to verify a composition of systems using these methods, one is forced to compute the composition of the original systems and then perform model reduction for the specific formulas on the (generally much larger) composite system.

For the logic CTL*, a super set of linear and branching temporal logics, strong bisimulation preserves the truth values of the standard satisfaction relation for all formulas [4],[11]. Strong bisimulation equivalence is often too strong to provide a significant reduction in the state size of the model. While this deficiency spawned the formula specific reductions described above, it also has lead to formula independent methods that achieve greater reduction at the price of preserving the truth values of a smaller class of formulas.

The formula independent methods of model reduction are typically based upon the algebraic equivalences derived from the work of Hoare [10] and Milner [16]. In [12], Kaivola and Valmari provide a method of "weak" model reduction for a nexttime-less linear temporal logic based upon failure equivalence [10]. As one might expect, the algorithm is worst case exponential. The paper [12] deals with state based models that are converted into event oriented models by labeling transitions with the changes they cause in the states (similar to [13]). Though this equivalence should work in a compositional setting, no parallel composition operators are considered. The labeling of transitions by state changes makes it unclear how one would define such a parallel composition operator. This is in contrast to the state-event setting of [9] where the separation of state values and event labels provides the standard event synchronization parallel composition operators. In [9] Graf and Loiseaux provide conditions under which abstractions preserving safety properties expressible in a fragment of the branching time μ-calculus are compositionally consistent. Their underlying model of state-event systems, which is equivalent to the State-Event Labeled Transition Systems (SELTS) used in this paper, permits synchronous products of systems. Their "strong" abstraction does not deal with fairness properties.

In our work we provide a method of "weak", compositionally consistent model reduction for state-event systems that preserves a class of safety and fairness properties related to systems' observed behaviors. The state-event equivalence relation we use for our form of formula independent model reduction is an extension of Milner's weak observation (bisimulation) equivalence. Kaivola and Valmari rejected weak observation equivalence for model reduction on the grounds that it did not necessarily preserve fairness properties due to its inability to distinguish divergences (infinite sequences of unobservable events). This is not a problem for logics such as Ostroff's RTTL which has the requirement that an (observable) *tick* of the global clock must occur infinitely often in any legal computation.

In the next section we use SELTS to model modules that can be combined via parallel composition operators to create new modules and systems. We also define a simple (real-time) state-event temporal logic that can be used for system specification and review the strong and weak state-event equivalences of [14]. Section 3 demonstrated how strong state-event equivalence can be used as the basis of a strong, compositionally consistent and computationally efficient model reduction technique for our entire logic. Section 4 obtains a weak model reduction technique for the subclass of state-event stuttering invariant formulas. While achieving greater reduction through the restriction of the formulas to be preserved, the reduced models of this section are still computable in polynomial time.

2 PRELIMINARIES

2.1 State-Event Labeled Transition Systems

SELTS extend Labeled Transition Systems (LTS) by adding a state output map. In our temporal logic setting the state output will be the set of atomic propositions satisfied by a state. While Kripke structures are generally used as the underlying model for temporal logic model checkers [6] and are ultimately the model we would employ in any model checking algorithms, considering structures that are extended by transition labels has two main benefits of incorporating system component timing information and synchronization information as described in the introduction.

Definition 1 *A* **State-Event Labeled Transition System** *(SELTS) is a 5-tuple* $\mathbb{Q} = \langle Q, \Sigma, R_\Sigma, q_0, P\rangle$ *where* Q *is an at most countable set of states,* Σ *is a finite set of elementary actions or events,* $R_\Sigma = \{\xrightarrow{\alpha} : \alpha \in \Sigma\}$ *is a set of binary relations on* Q*,* $q_0 \in Q$ *is the initial state and* $P : Q \to \mathcal{P}(AP)$ *is the state output map.*

In the above definition if $\alpha \in \Sigma$ and $q, q' \in Q$, then $q\xrightarrow{\alpha}q'$ means that the SELTS can move from state q to q' by executing elementary action α. Any transition relation $\xrightarrow{\alpha} \in R_\Sigma$ can be viewed as a function $\alpha^{\mathbb{Q}} : Q \to \mathcal{P}(Q)$, where $\mathcal{P}(Q)$ is the power set of Q. The function $\alpha^{\mathbb{Q}}$ maps q to the set of states reachable from q via a single α transition in the SELTS $\mathbb{Q}$. When the SELTS to which we are referring is obvious from the context, we will simply write $\alpha(q)$. For simplicity we assume $Q \neq \emptyset$ and $|Q|$ is finite. This assumption is justified since we intend to perform model reduction for finite state model checking. $AP, AP_1, AP_2, \ldots$ represent sets of atomic propositions. The state output map takes each state to the set of atomic propositions satisfied by the state (ie. $P(q) \subseteq AP$).

We now define a synchronous composition operator to provide a mechanism for constructing large systems consisting of interacting subsystems. In this version of the paper we deal with a strictly event base synchronization operator. The synchronous composition operator we use here is a straight forward extension to the SELTS setting of the parallel composition operator of [16].

Definition 2 *Given SELTS,* $\mathbb{Q}_i = \langle Q_i, \Sigma_i, R^i_\Sigma, q_{i0}, P_i\rangle$ *with* $P_i : Q_i \to \mathcal{P}(AP_i)$ *for* $i = 1, 2$ *such that* $AP_1 \cap AP_2 = \emptyset$ *and a set of events* $\Sigma_s \subseteq \Sigma_1 \cap \Sigma_2$*,* the Σ_s-synchronous product of $\mathbb{Q}_1$ and $\mathbb{Q}_2$ *is given by:* $\mathbb{Q}_1|[\Sigma_s]|\mathbb{Q}_2 := \langle Q_1 \times Q_2, \Sigma_1 \cup \Sigma_2, R_{\Sigma_1\cup\Sigma_2}, (q_{10}, q_{20}), P\rangle$*, where* $P : Q_1 \times Q_2 \to \mathcal{P}(AP_1 \cup AP_2)$ *is defined by* $P((q_1, q_2)) = P_1(q_1) \cup P_2(q_2)$ *and the elements of* $R_{\Sigma_1\cup\Sigma_2} = \{\xrightarrow{\alpha} : \alpha \in \Sigma_1\cup\Sigma_2\}$ *are binary relations over* $Q_1\times Q_2$ *defined as follows:* $(q_1, q_2)\xrightarrow{\alpha}(q_1', q_2')$ *iff*

(i) $\alpha \in \Sigma_s$*, and* $q_i\xrightarrow{\alpha}q_i'$ *in* $\mathbb{Q}_i$ *for* $i = 1, 2$*, or*

(ii) $\alpha \notin \Sigma_s$, $q_1 \xrightarrow{\alpha} q_1'$ *in* $\mathbb{Q}_1$ *and* $q_2 = q_2'$, *or*
(iii) $\alpha \notin \Sigma_s$, $q_2 \xrightarrow{\alpha} q_2'$ *in* $\mathbb{Q}_2$ *and* $q_1 = q_1'$.

We now introduce some notation to aid in our discussion of generated and observed state-event sequences. We define $\Sigma_- := \Sigma \cup \{-\}$ and $S := Q \times \Sigma_-$. For $s = (q, \alpha) \in S$, in addition to the set of atomic proposition found in $P(q)$ we associate the atomic proposition $\eta = \alpha$. We refer to η as the (next) transition variable. The computations of the SELTS $\mathbb{Q}$ will then be a subset of the union of the set of all finite, non-empty, state-event sequences S^+, and the set of all infinite state-event sequences S^ω. We also introduce the notation $|\sigma|$, which for $\sigma = s_0 s_1 s_2 \dots s_n \in S^+$ is defined as $|\sigma| = n$ and for $\sigma \in S^\omega$, $|\sigma| = \omega$.

Definition 3 *Given a SELTS* $\mathbb{Q}$*, the set of* **computations** *of* $\mathbb{Q}$*, denoted* $\mathcal{M}(\mathbb{Q})$*, is the largest subset of* $S^+ \cup S^\omega$ *such that for all* $\sigma \in \mathcal{M}(\mathbb{Q})$, $\sigma = s_0 s_1 \dots s_n = (q_0, \alpha_0)(q_1, \alpha_1) \dots (q_n, -) \in S^+$ *or* $\sigma = s_0 s_1 \dots = (q_0, \alpha_0)(q_1, \alpha_1) \dots \in S^\omega$ *and*
(i) Initialization: q_0 *is the initial state of* $\mathbb{Q}$.
(ii) Succession: $0 \leq i < |\sigma|$ *implies* $\alpha_i \in \Sigma$ *and* $q_{i+1} \in \alpha^{\mathbb{Q}}(q_i)$ *(ie.* $q_i \xrightarrow{\alpha}_i q_{i+1}$ *in* $\mathbb{Q}$*).*
(iii) Diligence: $\alpha_i = -$ *iff* $i = |\sigma|$ *and for all* $\alpha \in \Sigma, \alpha^{\mathbb{Q}}(q_i) = \emptyset$.

In Definition 3 conditions (i) and (ii) guarantee, respectively, that the computation starts in the system's initial state and the change from one state to the next via the given event is possible in $\mathbb{Q}$. (iii) states that the only finite sequences in $\mathcal{M}(\mathbb{Q})$ terminate in a state where no transitions are possible and hence the final "event" of the state-event sequence is denoted by $-$. (iii) differs from [15] since there is no idling transition in our setting. We allow finite sequences of states to be computations and modify our definition of temporal semantics accordingly [1].

2.2 Temporal Logic of State-Event Sequences

We now give a brief summary of temporal logic and refer the reader to [15],[17],[1] for the full details. Following [17], the state-event sequences defined above will play the role of state sequences in [15]. RTTL, as an example of a state-event temporal logic, is based upon Manna-Pnueli temporal logic with additional proof rules for dealing with real-time (*tick* event) properties. To allow us to express simple real-time properties we add a bounded until operator.

State formulas and *state-event formulas* are arbitrary boolean combinations of atomic predicates such that state formulas do not include any transition predicates such as $\eta = \alpha$ while state-event formulas *may* include such predicates. Neither contain any temporal operators. For a state formula F_s and a state q, we use the standard inductive definition of satisfaction and write $q \models F_s$ when F_s is true in state q. Similarly the definition of satisfaction can be extended to any state-event pair $s \in S$ and any state-event formula F_{se}.

In the following inductive definition of temporal state-event formulas we will consider an arbitrary (possibly finite) state-event sequence $\sigma = s_0 s_1 \dots = (q_0, \alpha_0)(q_1, \alpha_1) \dots$. Henceforth σ^k will be used to denote the k-shifted suffix of σ, $s_k s_{k+1} \dots = (q_k, \alpha_k)(q_{k+1}, \alpha_{k+1}) \dots$, when it exists (ie. when $|\sigma| \geq k$). For each $\alpha \in \Sigma$ we use the notation $\#\alpha(\sigma, i)$ to denote the number of α transitions that occur between the q_0 and q_i of the state-event sequence σ. If $|\sigma| < i$ then $\#\alpha(\sigma, i)$ is undefined.

Definition 4 *For temporal formulas F, F_1, F_2 and state-event sequence σ, the* **satisfaction relation** *is defined as follows:*

- *If $F \in AP$ is an atomic predicate, then $\sigma \models F$ iff $s_0 \models F$ (ie. $F \in P(q_0)$)*
- *If $F := \eta = \alpha$, then $\sigma \models F$ iff $(\alpha_0 = \alpha)$*
- *$\sigma \models F_1 \vee F2$ iff $\sigma \models F_1$ or $\sigma \models F_2$*
- *$\sigma \models \neg F$ iff $\sigma \not\models F$*
- *$\sigma \models \bigcirc F$ iff σ^1 exists and $\sigma^1 \models F$*
- *$\sigma \models F_1 \mathcal{U} F_2$ iff $\sigma \models F_2$ or $\exists k > 0$ such that σ^k is defined, $\sigma^k \models F_2$ and $\forall i, 0 \leq i < k, \sigma^i \models F_1$.*
- *$\sigma \models F_1 \mathcal{U}^{\alpha}_{[l,u]} F_2$ iff $\sigma \models F_2$ or $\exists k > 0$ such that σ^k is defined, $\sigma^k \models F_2$ and $\forall i, 0 \leq i < k, \sigma^i \models F_1$ and $l \leq \#\alpha(\sigma, k) \leq u$.*

The "next" operator $\bigcirc$ and "until" operator $\mathcal{U}$ are typically used to define additional operators (eg. the "eventually" operator $\Diamond F$, denotes $(true)\mathcal{U}F$, and the "henceforth" operator $\Box F$, denotes $\neg\Diamond\neg F$). As an example temporal formula, consider $F := \Box\bigcirc true$. F is satisfied only by those σ such that $|\sigma| = \omega$. The $\mathcal{U}^{\alpha}_{[l,u]}$ operator is just the until operator subject to the restriction that for a formula $F_1\mathcal{U}^{\alpha}_{[l,u]}F_2$, F_2 must become true after the lth occurrence of α and before the $(u + 1)$th occurrence of α. In systems in which time is represented by discrete *tick* events the $\mathcal{U}^{tick}_{[l,u]}$ operator can be used to specify that a system meets hard time bounds. For example, any system satisfying the formula $(true)\mathcal{U}^{tick}_{[0,2]}(\eta = \beta)$ will produce a β event before 3 time units have passed.

Definition 5 *Given a SELTS $\mathbb{Q}$ and a temporal formula F, we say that F is* $\mathbb{Q}$-valid, *written $\mathbb{Q} \models F$, iff for all $\sigma \in \mathcal{M}(\mathbb{Q})$, $\sigma \models F$.*

Fairness

Typically when a given transition structure is used as the model for a system, a designer specifies some fairness constraints which a computation must satisfy if it is to be considered a "legal" computation of the system. For example, all systems in RTTL have the fairness constraint that the *tick* event must occur infinitely often ($\Box\Diamond(\eta = tick)$); the system must not stop the clock or permit an infinite number of non-*tick* transitions to occur between successive clock *tick*s. Given a specification as a temporal formula F, one is not interested in verifying that *all* the computations of the transition structure satisfy F but rather that all the *legal* computations satisfy F. That is $\mathbb{Q} \models \neg F_{fair} \vee F$, where F_{fair} is the conjunction of all fairness constraints. This method assumes that the set of legal computations considered is non-empty.

2.3 State-Event Equivalence

In this subsection we will summarize the results of [14]. State observations are provided by the state output map $P : Q \rightarrow \mathcal{P}(AP)$. We assume that any change in the truth values of the atomic predicates can be seen by an external observer, thus two states $q, q' \in Q$ produce the same output observation precisely when $P(q) = P(q')$. Denote the set of all equivalence relations on Q by $Eq(Q)$. Any state output map $P : Q \rightarrow \mathcal{P}(AP)$ induces an equivalence relation $\ker(P) \in Eq(Q)$, the equivalence kernel of P, given by $(q_1, q_2) \in \ker(P)$ if and only if $P(q_1) = P(q_2)$. $Eq(Q)$ becomes a complete lattice under the operations of relational intersection $\wedge$ and union of relational products $\vee$.

When each $\theta \in Eq(Q)$ is associated with the partition of Q corresponding to the cells of θ, the lattice of equivalence relations is isomorphic to the poset lattice of partitions of Q with the partial order $\theta_1 \leq \theta_2$ iff each cell of θ_1 is a subset of a cell of θ_2. Thus we can talk interchangeably about equivalence relations and partitions. When talking about partitions $\theta_1 \wedge \theta_2 \in Eq(Q)$ $(\theta_1 \vee \theta_2)$ is the coarsest (finest) partition finer (coarser) than both θ_1 and θ_2. We will denote the trivial partitions $\{\{q\} : q \in Q\} = \inf(Eq(Q))$ and $\{Q\} = \sup(Eq(Q))$ by Δ and ∇ respectively.

Strong State-Event Equivalence

In the strong state-event observational setting it is not only the state output sequences that are important, but also the connecting events (relations). This is illustrated by the following three sequences and their images under a state output map $P : Q \to \mathcal{P}(AP)$. Here $r_1, r_2 \in \mathcal{P}(AP)$.

$$\left.\begin{array}{l} q_{11}\xrightarrow{\tau}q_{12}\xrightarrow{\alpha}q_{13}\\ q_{21}\xrightarrow{\alpha}q_{22}\xrightarrow{\tau}q_{23}\\ q_{31}\xrightarrow{\tau}q_{32}\xrightarrow{\alpha}q_{33}\end{array}\right\}\overset{P}{\longmapsto}\left\{\begin{array}{l} r_1\xrightarrow{\tau}r_1\xrightarrow{\alpha}r_2\\ r_1\xrightarrow{\alpha}r_2\xrightarrow{\tau}r_2\\ r_1\xrightarrow{\tau}r_2\xrightarrow{\alpha}r_2\end{array}\right. \tag{1}$$

Later τ will be used to denote unobservable events but for now we assume that all τ transitions are observable. The first output sequence differs from the other two in the second state output while the second and third differ in the ordering of their connecting relations or "events". Thus no two of these sequences of states and connecting events produce identical output sequences. The class of *compatible partitions* plays the role of congruences for nondeterministic relations.

Definition 6 *(cf.[3]) Given a SELTS $\mathbb{Q} = \langle Q, \Sigma, R_\Sigma, q_0, P\rangle$, a partition $\theta \in Eq(Q)$ is a* compatible partition *for $\mathbb{Q}$ if for all $\alpha \in \Sigma$, whenever q, q' are in the same partition block (cell) C_i, then for any block C_j of θ, $\alpha(q) \cap C_j \neq \emptyset$ iff $\alpha(q') \cap C_j \neq \emptyset$. The set of all compatible partitions for the SELTS $\mathbb{Q}$ will be denoted by $CP(\mathbb{Q})$.*

From the above definition we see that for $\theta \in CP(\mathbb{Q})$ if $(q, q') \in \theta$ and $q\xrightarrow{\alpha}q_1$ then there exists q'_1 such that $q'\xrightarrow{\alpha}q'_1$ and $(q_1, q'_1) \in \theta$. In [14] it is shown that $CP(\mathbb{Q})$ is not closed under the $\wedge$ operation of $Eq(Q)$ but is closed under the $\vee$ operator of $Eq(Q)$. Thus for any $\mathcal{F} \subseteq CP(\mathbb{Q})$, there is a unique supremal element $\theta^* := \sup(\mathcal{F})$ and $\theta^* \in CP(\mathbb{Q})$. We are now in a position to characterize a strong state-event observer for any given SELTS.

Definition 7 *Given a SELTS $\mathbb{Q} = \langle Q, \Sigma, R_\Sigma, q_0, P\rangle$, the* strong state-event observer*, $\theta_s(\mathbb{Q})$ is defined to be $\theta_s(\mathbb{Q}) = \sup\{\theta \in CP(\mathbb{Q}) : \theta \leq \ker(P)\}$.*

When $\mathbb{Q}$ is clear from the context we will simply write θ_s for $\theta_s(\mathbb{Q})$. θ_s is the coarsest compatible partition of $\mathbb{Q}$ that is finer than the equivalence kernel of P. Thus θ_s represents the minimum information one needs about the current state to be able to predict all possible future state and event outputs.

In [3], Bolognesi *et al.* provide an $O(m \log n)$ algorithm, where m is the size of R_Σ (the number of related pairs) and $n = |Q|$, for computing Milner's strong observation equivalence $\sim$ for finite state Labeled Transition Systems. The algorithm is based upon the RCP (Relational Coarsest Partition problem) with an initial partition equal to $\nabla = \{\{Q\}\}$. This algorithm is easily adapted to computing θ_s without any change in complexity (assuming $\ker(P)$ is provided) by allowing the initial partition for the RCP to be $\ker(P)$. This close connection with $\sim$ leads us to write $q \sim_{se} q'$ when $(q, q') \in \theta_s$ and say that q is strong state-event observation equivalent to q'.

Like congruences, compatible partitions can be used to construct quotient systems.

Definition 8 *Given a SELTS $\mathbb{Q} := \langle Q, \Sigma, R_\Sigma, q_0, P\rangle$, for $\theta \in CP(\mathbb{Q})$ such that $\theta \leq \ker(P)$, we define the* quotient system *of $\mathbb{Q}$ by θ, $\mathbb{Q}/\theta$, as follows: $\mathbb{Q}/\theta := \langle Q/\theta, \Sigma, R_\Sigma/\theta, q_0/\theta, P_\theta\rangle$. Here q_0/θ denotes the cell of the partition θ containing q_0 and Q/θ denotes the set of all cells of θ. For $\alpha \in \Sigma$, the transition relations of R_Σ/θ are defined as $\alpha^{\mathbb{Q}/\theta}(q/\theta) = \alpha^{\mathbb{Q}}(q)/\theta = \{q_1/\theta \in Q/\theta : q_1 \in \alpha^{\mathbb{Q}}(q)\}$. $P_\theta : Q/\theta \to \mathcal{P}(AP)$ is the unique map such that $P_\theta \circ \theta(q) = P(q)$.*

Strong state-event observation equivalence can be extended to a relation $\sim_{se}$ between two SELTS having disjoint state sets. This is done by forming the union of the transition systems underlying LTS and the union of the original systems' state output maps. The two SELTS are then strongly state-event equivalent iff their initial states are strongly state-event observationally equivalent in the union system.

As in [1], we obtain the result that when $\mathbb{Q}$ is reachable, $\mathbb{Q}/\theta_s$ is the unique (up to isomorphism) minimal state SELTS for which $\mathbb{Q} \sim_{se} \mathbb{Q}/\theta_s$. The equivalence of a system with its quotient system together with the following result regarding the synchronous composition of equivalent systems provides us with the means for performing strong, compositionally consistent, model reduction.

Lemma 1 *Given SELTS $\mathbb{Q}_i, i = 1, 2$ and $\mathbb{R}_i, i = 1, 2$. If $\mathbb{Q}_i \sim_{se} \mathbb{R}_i, i = 1, 2$ then for all Σ_s such that $|[\Sigma_s]|$ is defined $(\mathbb{Q}_1|[\Sigma_s]|\mathbb{Q}_2) \sim_{se} (\mathbb{R}_1|[\Sigma_s]|\mathbb{R}_2)$.*

Weak State-Event Equivalence

Many Discrete Event Systems are event- rather than time-driven. In this case what is important is the sequence of changes in the outputs, ignoring intermediate states and events that do not generate any new outputs. Before applying this point of view in our state event setting, we review how it is applied in the event setting of Milner's weak observation equivalence. Again, (event) observation equivalence becomes the special case of our setting in which $\ker(P) = \nabla$.

Consider a LTS $\mathbb{Q} := \langle Q, \Sigma, R_\Sigma, q_0\rangle$. In the style of [3], we assume there is a "silent event" $\tau \in \Sigma$ that represents unobservable actions. We then define the set of observable actions to be $\Sigma_o := \Sigma - \{\tau\}$. This leads to some new relations on Q. Letting ϵ represent the empty string (over Σ), we say that q moves unobservably (from an event perspective) to q', written $q\overset{\epsilon}{\Rightarrow}q'$, iff there exist $q_0, q_1, \ldots, q_n \in Q$, $n \geq 0$, such that $q = q_0\xrightarrow{\tau}q_1\xrightarrow{\tau}\ldots\xrightarrow{\tau}q_{n-1}\xrightarrow{\tau}q_n = q'$. By convention, for any $q \in Q$, $q\overset{\epsilon}{\Rightarrow}q$. For $\alpha \in \Sigma_o$ we can then say that q moves to q' while producing event α, written $q\overset{\alpha}{\Rightarrow}q'$, iff there exist $q_1, q_2 \in Q$ such that $q\overset{\epsilon}{\Rightarrow}q_1\xrightarrow{\alpha}q_2\overset{\epsilon}{\Rightarrow}q'$.

In the weakly observable setting the actions $q\xrightarrow{\alpha}q'$ and $q\overset{\alpha}{\Rightarrow}q'$ are indistinguishable since both produce the single event output α. For a given $\mathbb{Q}$, these double arrow relations can be used to define a new transition system, $\mathbb{Q}' := \langle Q, \Sigma, R'_\Sigma, q_0\rangle$, where R'_Σ is defined as follows. For all $\alpha \in \Sigma_o$, $\alpha^{\mathbb{Q}'}(q) = \{q_1 \in Q : q\overset{\alpha}{\Rightarrow}q_1 \text{ in } \mathbb{Q}\}$ and $\tau^{\mathbb{Q}'}(q) = \{q_1 \in Q : q\overset{\epsilon}{\Rightarrow}q_1 \text{ in } \mathbb{Q}\}$.

Two states are weakly observation equivalent in $\mathbb{Q}$, written $q \approx q'$, iff the states are strongly observation equivalent ($q \sim q'$) in $\mathbb{Q}'$ so $\approx := \sup(CP(\mathbb{Q}'))$. Then $\approx$ represents the minimum information you need about Q to know what choices of future observable events are possible.

We now generalize weak observation equivalence to our state-event setting. Given a SELTS $\mathbb{Q} := \langle Q, \Sigma, R_\Sigma, q_0, P\rangle$, assume that the special event τ represents unobservable events. When a τ transition occurs, it does not produce an output event, though it may cause a change in the state output. For instance, if $q\xrightarrow{\tau}q'$ and $P(q) = P(q')$ then there is no noticeable change in the system output. If, on the other hand, $q\xrightarrow{\tau}q'$ and $P(q) \neq P(q')$ then although no event is seen to take place, a change in state output takes place when τ occurs. This leads us to define, for the given SELTS $\mathbb{Q}$, an unobservable move from q to q', written $q \Rightarrow_{se} q'$ iff there exist $q_0, q_1, \ldots, q_n \in Q$, $n \geq 0$, such that $q = q_0\xrightarrow{\tau}q_1\xrightarrow{\tau}\ldots\xrightarrow{\tau}q_{n-1}\xrightarrow{\tau}q_n = q'$ and for all $i = 0, 1, \ldots, n$ $P(q_i) = P(q) = P(q')$

Thus the relation $\Rightarrow_{se}$ is the transitive closure of the τ relation within each cell of $\ker(P)$. By convention $q \Rightarrow_{se} q$ always holds. While the $\Rightarrow_{se}$ relation captures a relation which is indistinguishable from the case when $q\xrightarrow{\tau}q'$ and $P(q) = P(q')$, we now wish to define a relation which captures both this case and the case when $q\xrightarrow{\tau}q'$ and $P(q) \neq P(q')$. We say that q moves to q' without an event output, written $q\overset{\epsilon}{\Rightarrow}_{se}q'$, iff $q = q'$, or there exist $q_1, q_2 \in Q$ such that $q \Rightarrow_{se} q_1\xrightarrow{\tau}q_2 \Rightarrow_{se} q'$. Note by definition $q\overset{\epsilon}{\Rightarrow}_{se}q$. The relation $\overset{\epsilon}{\Rightarrow}_{se}$ is the transitive closure of $\xrightarrow{\tau}$ subject to the restriction that at most one boundary of the partition $\ker(P)$ is crossed. If $q\overset{\epsilon}{\Rightarrow}_{se}q'$, then no output events are generated and there is at most one change in the state output.

We now define a relation similar to $\overset{\epsilon}{\Rightarrow}_{se}$ except that it produces exactly one event output. For $\alpha \in \Sigma_o$, we say that q moves to q' producing event output α, written $q\overset{\alpha}{\Rightarrow}_{se}q'$ iff there exist $q_1, q_2 \in Q$ such that $q \Rightarrow_{se} q_1\xrightarrow{\alpha}q_2 \Rightarrow_{se} q'$. We emphasize that if a boundary of $\ker(P)$ is crossed when $q\overset{\alpha}{\Rightarrow}_{se}q'$, then it is only crossed by the α transition.

Consider the state event sequences (1) at the start of this subsection from the point of view that only output (observable) events and changes in the state output are important. The first two sequences are indistinguishable. In both sequences the event α and the state output change from r_1 to r_2 occur simultaneously. Hence $q_{11}\overset{\alpha}{\Rightarrow}_{se}q_{13}$ and $q_{21}\overset{\alpha}{\Rightarrow}_{se}q_{23}$ and in both cases at the output it appears as $r_1\xrightarrow{\alpha}r_2$. In the case of the third string, the state output changes with the unobservable transition τ and *then* the event α occurs. Thus $q_{31}\overset{\epsilon}{\Rightarrow}_{se}q_{32}\overset{\alpha}{\Rightarrow}_{se}q_{33}$ but not $q_{31}\overset{\alpha}{\Rightarrow}_{se}q_{33}$ and so at the outputs the third sequence appears as $r_1\xrightarrow{\tau}r_2\xrightarrow{\alpha}r_2$. From a control point of view it is important that an observer distinguish the first two sequences from the third. Assume that r_2 is a bad state output we wish to avoid and α is a controllable event that can be disabled as in [20]. Disabling α prevents state output r_2 from occurring in the first two sequences of (1) but not in the third sequence!

With the above examples in mind, we are ready to define weak state-event observers.

Definition 9 *Given a SELTS* $\mathbb{Q} = \langle Q, \Sigma, R_\Sigma, q_0, P\rangle$, *the* weak state-event observer *is given by* $\theta_w(\mathbb{Q}) := \sup\{\theta \in CP(\mathbb{Q}'_{se}) : \theta \leq \ker(P)\}$. *Here* $\mathbb{Q}'_{se} := \langle Q, \Sigma, R'_\Sigma, q_0, P\rangle$ *where* R'_Σ *is defined as follows. For all* $\alpha \in \Sigma_o$, $q\xrightarrow{\alpha}q'$ *in* $\mathbb{Q}'_{se}$ *iff* $q\overset{\alpha}{\Rightarrow}_{se}q'$ *in* $\mathbb{Q}$ *and* $q\xrightarrow{\tau}q'$ *in* $\mathbb{Q}'_{se}$ *iff* $q\overset{\epsilon}{\Rightarrow}_{se}q'$ *in* $\mathbb{Q}$.

θ_w always exists and is unique. Note that in $\mathbb{Q}'_{se}$ the transition relations are dependent upon the SELTS state output map P so θ_w is not just Milner's observation equivalence with a different initial partition. When $\ker(P) = \nabla$ then θ_w is $\approx$, Milner's weak observation equivalence, since $\mathbb{Q}'_{se}$ becomes $\mathbb{Q}'$. Similar to strong state-event equivalence, when $(q, q') \in \theta_w$ for a given $\mathbb{Q}$, we will write $q \approx_{se} q'$, read "q is weak state-event observationally equivalent to q'". The $O(n^3)$ algorithm ($n = |Q|$) for computing Milner's weak observation equivalence of finite state LTS given in [3] can be easily adapted to provide an $O(n^3)$ algorithm for θ_w.

θ_w is the coarsest compatible partition of $\mathbb{Q}'_{se}$ that is finer than the equivalence kernel of P. Thus for $(q, q') \in \theta_w$ we have $P(q) = P(q')$ so q and q' produce the same current state output. Also q' can generate state and event outputs that are indistinguishable from those produced from q. θ_w represents the minimum information one needs about the current state to be able to predict all possible future changes in state and future event outputs.

Since the weak state-event observer for a SELTS $\mathbb{Q}$ is just the strong state-event observer for $\mathbb{Q}'_{se}$, we are able to state similar results about *weak quotient systems*. In defining weak quotient systems we use the intuition that in the weakly observable setting the actions $q\xrightarrow{\alpha}q'$ and $q\overset{\alpha}{\Rightarrow}_{se}q'$ are indistinguishable.

Definition 10 *Given an SELTS* $\mathbb{Q} := \langle Q, \Sigma, R_\Sigma, q_0, P\rangle$, *for* $\theta \in CP(\mathbb{Q}'_{se})$ *such that* $\theta \leq \ker(P)$, $\mathbb{Q}//\theta := \mathbb{Q}'_{se}/\theta$ *is the* weak quotient system of $\mathbb{Q}$ by θ.

Again we can extend weak state-event observation equivalence to a relation $\approx_{se}$ between SELTS by forming the union of disjoint SELTS. When $\mathbb{Q}$ is reachable, $\mathbb{Q}//\theta_w$ is a minimal state SELTS such that $\mathbb{Q}//\theta_\omega \approx_{se} \mathbb{Q}$. Since weak observation equivalence ignores differences resulting from unobservable τ transitions, below in the weak state-event version of Lemma 1 we require that τ is not part of the synchronization set.

Lemma 2 *Given SELTS* $\mathbb{Q}_i, i = 1, 2$ *and* $\mathbb{R}_i, i = 1, 2$. *If* $\mathbb{Q}_i \approx_{se} \mathbb{R}_i, i = 1, 2$ *then for all* Σ_s *such that* $\tau \notin \Sigma_s$ *we have* $(\mathbb{Q}_1|[\Sigma_s]|\mathbb{Q}_2) \approx_{se} (\mathbb{R}_1|[\Sigma_s]|\mathbb{R}_2)$.

3 STRONG STATE-EVENT MODEL REDUCTION

In this section we assume that while we have perfect event information (all events including τ events are observable), only partial state information is provide via the state output map. The main result of this section is that strongly state-event equivalent systems satisfy the same temporal formulas and hence we can use a systems strong state-event quotient system to verify system properties. The compositional consistency of this model reduction technique then follows immediately from the fact that strong state-event equivalence is a precongruence for the SELTS $|[\Sigma_s]|$ synchronous composition operator. While the results obtained in this section follow easily from the truth preserving properties of strong bisimulation equivalence, the technique employed in this section will be utilized in following section on weak state-event model reduction.

Unless stated otherwise, we henceforth assume that we are dealing with a SELTS $\mathbb{Q} = \langle Q, \Sigma, R_\Sigma, q_0, P\rangle$ where state output map $P : Q \to \mathcal{P}(AP)$, and AP is the set of atomic predicates of interest. Given a computation σ, the *strongly observed computation* generated by σ is given by applying P to the state of each state event pair in the computation. This provides a map from sequences over $Q \times \Sigma_-$ to sequences over $\mathcal{P}(AP) \times \Sigma_-$, $P^\sim : Q \times \Sigma_- \to \mathcal{P}(AP) \times \Sigma_-$. That is, $P^\sim((q_0, \alpha_0)(q_1, \alpha_1) \ldots (q_n, \alpha_n) \ldots) = (P(q_0), \alpha_0)(P(q_1), \alpha_1) \ldots (P(q_n), \alpha_n) \ldots$. For C, a set of computations, define $P^\sim(C) := \{P^\sim(\sigma) : \sigma \in C\}$.

Lemma 3 *Let* $\mathbb{Q}_i = \langle Q_i, \Sigma, R^i_\Sigma, q_{i0}, P_i\rangle, i = 1, 2$ *be SELTS. If* $\mathbb{Q}_1 \sim_{se} \mathbb{Q}_2$ *then* $P_1^\sim(\mathcal{M}(\mathbb{Q}_1)) = P_2^\sim(\mathcal{M}(\mathbb{Q}_2))$

The systems in Figure 1 demonstrate that the converse Lemma 3 is false. The transition systems are shown with the state outputs generated by their respective state output maps P_1 and P_2 next to each state. The initial states of the two transition systems are marked by entering arrows. In this case $P_1^\sim(\mathcal{M}(\mathbb{Q}_1)) = P_2^\sim(\mathcal{M}(\mathbb{Q}_2)) = \{r_1\xrightarrow{\alpha}r_1\xrightarrow{\beta}r_2\xrightarrow{\delta}r_2\xrightarrow{\delta}\ldots, r_1\xrightarrow{\alpha}r_1\xrightarrow{\beta}r_2\xrightarrow{\gamma}r_2\xrightarrow{\gamma}\ldots\}$

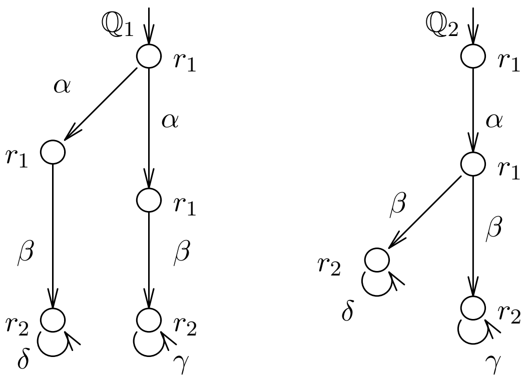

Figure 1 Counter example to converse of Lemma 3.

but, as can be easily verified, $\mathbb{Q}_1 \not\sim_{se} \mathbb{Q}_2$. By extending Hoare's failure equivalence to a state-event failure equivalence in a manner similar to the way that (event) observation equivalence was extend to state-event observation equivalence, one obtains an equivalence which relates the two systems of Figure 1. Unfortunately the computation of failure equivalence is PSPACE-complete making it unlikely that an efficient algorithm could be found to compute any extension to the state-event setting. On the other hand strong state-event equivalence is $O(n \log m)$ making state-event equivalence preferable as a practical model reduction technique.

As an immediate consequence of Lemma 3, we obtain the following result.

Theorem 1 *Given two SELTS as above, if $\mathbb{Q}_1 \sim_{se} \mathbb{Q}_2$ then for any temporal formula F, we have $\mathbb{Q}_1 \models F$ iff $\mathbb{Q}_2 \models F$.*

The above theorem allows us to use a system's strong state-event quotient system to reason about the state output and event behavior of the system since $\mathbb{Q} \sim_{se} \mathbb{Q}/\theta_s$. Lemma 1 together with Theorem 1 then guarantees that strong state-event equivalence can be used for compositionally consistent model reduction of SELTS for all formulas in state-event temporal logic.

4 WEAK STATE-EVENT MODEL REDUCTION

We now turn our attention to the case with only partial event observations in addition to the partial state observations provided by the state output map. We assume that all unobservable transitions are labeled by τ. We want to reason about the sequences of observed events and changes in state output. To this end we define a projection from computations to weakly observed computations similar to the strong projection of the previous section. This time we delete a state-event pair from the strongly observed computation if the event is an unobservable τ transition and the state output remains unchanged in the next state. Since weak state-event equivalence ignores system information regarding sequences of unobservable events that do not cause state changes, the equivalence can only be used for model reduction with a restricted set of temporal formulas. This restricted class, which we will call the class of State-Event Stuttering Invariant (SESI) formulas, is characterized as those formulas that are satisfied by a computation iff the projected computation satisfies the formula. We identify a set of SESI formulas, including some formulas making use of immediate operators ($\bigcirc, \eta =$). The main result of the section states that weakly state-event equivalent systems satisfy the same subset of SESI formulas.

4.1 Weakly Observed Computations and Weak Satisfaction

In [15] the authors use a state-based projection operator to develop a state-only version of weak satisfaction. They define the *reduced behavior* of a computation σ via a two step process that amounts to first applying $P^{\sim}$, the strong computation projection of the previous section, and then replacing uninterrupted sequences of identical "states" with a single copy of the state. In our case we are dealing with sequences of state-event pairs rather than just sequences of states. We cannot simply apply $P^{\sim}$ and then replace subsequences of uninterrupted state-event pairs by a single state-event pair since in this case important information relating state changes and event observations would be lost. Consider the three state-event sequences shown below where *tick* is the event representing the passage of one second on the global clock.

$$\begin{array}{l}
(q_0, \tau)(q_0, \tau)(q_0, \mathit{tick})(q_0, \alpha)(q_1, \mathit{tick}) \ldots \\
(q_0, \tau)(q_0, \mathit{tick})(q_0, \mathit{tick})(q_0, \alpha)(q_1, \mathit{tick}) \ldots \\
(q_0, \mathit{tick})(q_0, \tau)(q_0, \mathit{tick})(q_0, \tau)(q_0, \alpha)(q_1, \mathit{tick}) \ldots
\end{array}$$

If we assume that the state output map is the identity map, then following [15] the first and second sequences would result in the same reduced computation: $(q_0, \tau)(q_0, \mathit{tick})(q_0, \alpha)(q_1, \mathit{tick}) \ldots$, while the third sequence is its own reduced computation. This would lead us to believe that in the first two cases the system delays for one second and then changes state from q_0 to q_1 via an α transition when, in fact, the second and third computations do not make the α transition for 2 seconds. While we want our projection operator to distinguish the first case from the other two, the second and third computations differ only by unobservable transitions that do not change the state output. Upon rewriting the three sequences in terms of the notation of weak state-event observation equivalence, the differences and similarities in observed behaviors become apparent:

$$\left.\begin{array}{c}
q_0 \overset{\tau}{\rightarrow} q_0 \overset{\tau}{\rightarrow} q_0 \overset{\mathit{tick}}{\rightarrow} q_0 \overset{\alpha}{\rightarrow} q_1 \overset{\mathit{tick}}{\rightarrow} \ldots \\
q_0 \overset{\tau}{\rightarrow} q_0 \overset{\mathit{tick}}{\rightarrow} q_0 \overset{\mathit{tick}}{\rightarrow} q_0 \overset{\alpha}{\rightarrow} q_1 \overset{\mathit{tick}}{\rightarrow} \ldots \\
q_0 \overset{\tau}{\rightarrow} q_0 \overset{\mathit{tick}}{\rightarrow} q_0 \overset{\tau}{\rightarrow} q_0 \overset{\mathit{tick}}{\rightarrow} q_0 \overset{\alpha}{\rightarrow} q_1 \overset{\mathit{tick}}{\rightarrow} \ldots
\end{array}\right\} \overset{P}{\mapsto} \left\{\begin{array}{c}
q_0 \overset{\mathit{tick}}{\Rightarrow}_{se} q_0 \overset{\alpha}{\Rightarrow}_{se} q_1 \overset{\mathit{tick}}{\Rightarrow}_{se} \cdots \\
q_0 \overset{\mathit{tick}}{\Rightarrow}_{se} q_0 \overset{\mathit{tick}}{\Rightarrow}_{se} q_0 \overset{\alpha}{\Rightarrow}_{se} q_1 \overset{\mathit{tick}}{\Rightarrow}_{se} \cdots \\
q_0 \overset{\mathit{tick}}{\Rightarrow}_{se} q_0 \overset{\mathit{tick}}{\Rightarrow}_{se} q_0 \overset{\alpha}{\Rightarrow}_{se} q_1 \overset{\mathit{tick}}{\Rightarrow}_{se} \cdots
\end{array}\right.$$

To an external observer the second and third computations would produce the same observed state-event sequence: $(q_0, \mathit{tick})(q_0, \mathit{tick})(q_0, \alpha)(q_1, \mathit{tick}) \ldots$. The projection defined below has the effect of replacing all the state-event pairs making up a $q_1 \overset{\alpha}{\Rightarrow}_{se}$ transition with a single state-event pair $q_1 \overset{\alpha}{\rightarrow}$.

Definition 11 *Given a SELTS $\mathbb{Q}$ with state output map $P : Q \rightarrow \mathcal{P}(AP)$ and $\sigma = q_0 \overset{\alpha_0}{\rightarrow} q_1 \overset{\alpha_1}{\rightarrow} \ldots$, $\sigma \in \mathcal{M}(\mathbb{Q})$, the* weakly observed behavior *of σ is denoted by $P^{\approx}(\sigma)$ and defined inductively as:*

$$\begin{array}{rcl}
P^{\approx}(q_0) & = & P(q_0) \\
P^{\approx}(q_0 \overset{\alpha_0}{\rightarrow} q_1 \overset{\alpha_1}{\rightarrow} \ldots q_n \overset{\alpha_n}{\rightarrow} q_{n+1}) & = & \left\{\begin{array}{ll}
P^{\approx}(q_0 \overset{\alpha_0}{\rightarrow} q_1 \overset{\alpha_1}{\rightarrow} \ldots q_n), & \text{if } \alpha_n = \tau \wedge P(q_n) = P(q_{n+1}) \\
P^{\approx}(q_0 \overset{\alpha_0}{\rightarrow} q_1 \overset{\alpha_1}{\rightarrow} \ldots q_n) \overset{\alpha_n}{\rightarrow} P(q_{n+1}), & \text{otherwise}
\end{array}\right.
\end{array}$$

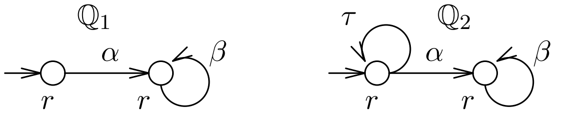

Figure 2 $\mathbb{Q}_1 \approx_{se} \mathbb{Q}_2$ but $P_1^{\approx}(\mathcal{M}(\mathbb{Q}_1)) \neq P_2^{\approx}(\mathcal{M}(\mathbb{Q}_2))$

For C a set of computations, we define $P^{\approx}(C) := \{P^{\approx}(\sigma) : \sigma \in C\}$.

Example 1 *In this example we consider the weak state-event observations generated by a SELTS with identity state output map $P := I_Q$ where $I_Q : Q \to Q$.*

$$\begin{aligned}
\sigma_1 &= (q_0,\tau)(q_0,\alpha)(q_0,\tau)(q_1,\tau)(q_1,\beta)(q_2,\alpha)\ldots = q_0\xrightarrow{\tau}q_0\xrightarrow{\alpha}q_0\xrightarrow{\tau}q_1\xrightarrow{\tau}q_1\xrightarrow{\beta}q_2\xrightarrow{\alpha}\ldots\\
P^{\approx}(\sigma_1) &= q_0\xrightarrow{\alpha}q_0\xrightarrow{\tau}q_1\xrightarrow{\beta}q_2\xrightarrow{\alpha}\ldots = (q_0,\alpha)(q_0,\tau)(q_1,\beta)(q_2,\alpha)\ldots\\
\sigma_2 &= (q_0,\tau)(q_0,\tau)(q_0,\tau)\ldots = q_0\xrightarrow{\tau}q_0\xrightarrow{\tau}q_0\xrightarrow{\tau}\ldots\\
P^{\approx}(\sigma_2) &= q_0 = (q_0,-)
\end{aligned}$$

In $P^{\approx}(\sigma_1)$ all the τ transitions are eliminated except for the $q_0\xrightarrow{\tau}q_1$ transition since this τ transition can be inferred from the external observer's observation of a state change from q_0 to q_1 without any observed event. In this case we say that τ is an *implicitly observed transition.* The computation σ_2 is initially observed to be in state q_0 and then produces no state change or event observations. This is reflected in $P^{\approx}(\sigma_2)$ as $(q_0,-)$, the observed state output with no defined transition. Thus an infinite state-event sequence can result in a finite weakly observed sequence.

As the basis of weak state-event model reduction, we would like to obtain a result similar to Lemma 3 which stated that strongly state-event equivalent systems result in the same set of strongly observed computations. In this case we have to be careful with our treatment of the unobservable transitions that are erased by the weak projection. Consider the two weakly state-event equivalent systems shown in Figure 2. Here $r \in \mathcal{P}(AP)$ is the same state output for all the systems' states. In this case $P_1^{\approx}(\mathcal{M}(\mathbb{Q}_1)) = \{r\xrightarrow{\alpha}r\xrightarrow{\beta}r\xrightarrow{\beta}r\xrightarrow{\beta}\ldots\}$ but $P_2^{\approx}(\mathcal{M}(\mathbb{Q}_1)) = \{r, r\xrightarrow{\alpha}r\xrightarrow{\beta}r\xrightarrow{\beta}r\xrightarrow{\beta}\ldots\}$. The above systems agree upon their trajectories that produces an infinite number of observations. It is the infinite sequence of unobservable τ's that $\mathbb{Q}_2$ can produce that causes the discrepancy. This observation is formalized in the following Lemma.

Lemma 4 *Given two SELTS, $\mathbb{Q}_i = \langle Q_i, \Sigma, R_\Sigma^i, q_{i0}, P_i\rangle$, where $P_i : Q_i \to \mathcal{P}(AP)$, $i = 1,2$, if $\mathbb{Q}_1 \approx_{se} \mathbb{Q}_2$ then $P^{\approx}(\mathcal{M}(\mathbb{Q}_1)) \cap (\mathcal{P}(AP)\times\Sigma)^\omega = P^{\approx}(\mathcal{M}(\mathbb{Q}_2)) \cap (\mathcal{P}(AP)\times\Sigma)^\omega$.*

The above lemma states that weakly state-event equivalent systems produce identical infinite sequences of observations, though equivalent systems may disagree on sequences that produce finite observations. In RTTL and the simplified real-time state-event logic presented here, the fairness constraint $\Box\Diamond(\eta = tick)$ guarantees that the clock *tick*s infinitely often in all legal computations (ie. all legal computations result in infinite sequences of observations). Thus if we can identify a subclass of formulas with truth values that are only dependent upon the observations a computation produces, the above lemma will allow us to use weak state-event equivalence to perform model reduction for the subclass.

As a first step towards obtaining a subclass of temporal formulas with truth values that are dependent upon the weakly observed computations, we will define weak satisfaction. While our main interest in introducing weak satisfaction is to obtain a subclass of formulas for weak state-event model reduction, weak satisfaction also provides a means of specifying behavior of weakly projected computations and hence of specifying the behavior of the system at its outputs or interface with other modules.

Definition 12 *Given a SELTS $\mathbb{Q}$ and a temporal formula F, a computation $\sigma \in \mathcal{M}(\mathbb{Q})$ is said to* **weakly satisfy** *F, written $\sigma \models_{\approx} F$, iff $P^{\approx}(\sigma) \models F$. The SELTS $\mathbb{Q}$ weakly satisfies F, written $\mathbb{Q} \models_{\approx} F$, iff $P^{\approx}(\mathcal{M}(\mathbb{Q})) \models F$.*

Example 2 *For σ_1 and σ_2 as in Example 1 we have $\sigma_1 \models_{\approx} \eta = \alpha \wedge q = q_0$ while $\sigma_2 \not\models_{\approx} \Box \bigcirc true$.*

In the case of σ_1 we are stating that the first observed action of the computation is an α transition that does not change the state output. In the case of σ_2 we are stating that the computation does not produce an infinite number of observations. A computation σ weakly satisfies $\Box \bigcirc true$ if the weak projection of the computation is an infinite sequence. Thus $\sigma \models_{\approx} \Box \bigcirc true$ becomes a concise way of say that σ produces an infinite number of observations.

Theorem 2 *Given two SELTS, if $\mathbb{Q}_1 \approx_{se} \mathbb{Q}_2$ then for any temporal formula F we have $\mathbb{Q}_1 \models_{\approx} \neg(\Box \Diamond \eta = tick) \vee F$ iff $\mathbb{Q}_2 \models_{\approx} \neg(\Box \Diamond \eta = tick) \vee F$.*

The implication of the above theorems is that weak state-event equivalence can be used to perform model reduction for any real-time state-event temporal logic formula provided the satisfaction relation of interest is weak satisfaction. In general we are interested in performing model reduction for the standard satisfaction relation $\models$. In the following subsection Theorem 2 will be the key to developing model reduction results for the subclass of SESI formulas under the standard satisfaction relation.

4.2 State-Event Stuttering Invariance and Model Reduction

We now consider those formulas with truth values that are robust with respect to unobservable τ transitions, ie. those formulas which have the property that for all computations σ

$$\sigma \models_{\approx} F \text{ iff } \sigma \models F \tag{2}$$

We call such formulas *State-Event Stuttering Invariant* (SESI). Equation (2) provides the link relating satisfaction to weak satisfaction that will be used to extend Theorem 2 to standard satisfaction of SESI formulas. We now try to identify some SESI formulas before providing a formal statement that allows us to build more general SESI formulas.

Let F_s be a state formula. Then $\sigma \models_{\approx} F_s$ iff $\sigma \models F_s$ since $P^{\approx}$ does not affect the value of the initial state output. The case for general state-event formulas is complicated by references to the (next) transition variable η. Considering Example 1 we see that $\sigma_1 \models (\eta = \tau)$ but $\sigma_1 \models_{\approx} (\eta = \alpha)$ (ie. the first transition of the computation is a τ transition but the first transition of the weakly observed computation is an α event). This difference results from the weak state-event projection operator deleting all τ transitions that do not cause any change in the state output. The formula $\diamond(\eta = \alpha)$ states that eventually an α transition occurs so clearly for any $\alpha \neq \tau$, $\sigma \models \diamond(\eta = \alpha)$ iff $\sigma \models_{\approx} \diamond(\eta = \alpha)$ since $P^{\approx}$ does not erase any non-τ transitions. With a similar argument one can also show that for $p \in AP$ and $\alpha \in \Sigma - \{\tau\}$, the formula $\Box[(\eta = \alpha) \rightarrow \bigcirc p]$, stating that in the state following an α transition p always holds, is SESI. Such "base" formulas can be used to build up more complex temporal formulas as outlined below.

Lemma 5 *Let σ be a computation and F, F_1, F_2 be SESI formulas. Then for $\alpha \in \Sigma - \{\tau\}, k \in \mathbb{N}$ we have $\neg F$, $F_1 \vee F_2$, $F_1 \mathcal{U} F_2$ and $F_1 \mathcal{U}^{\alpha}_{[l,u]} F_2$ are all SESI formulas.*

From the above discussion we see that all *non-immediate formulas*, formulas composed solely of state predicates together with the $\vee, \wedge, \mathcal{U}, \mathcal{U}^{\alpha}_{[l,u]}$ operators (ie. that do not contain the next operator $\bigcirc$ or next transition variable η) are SESI. Additionally, a formula of the form $\Box \diamond (\eta = tick)$ is SESI since $\diamond(\eta = tick)$ is SESI and $\Box F = \neg \diamond F$. We can now extend Theorem 2 to provide results about $\models$ for formulas that belong to the subclass of SESI formulas.

Theorem 3 *Let F be an SESI formula. If $\mathbb{Q}_1, \mathbb{Q}_2$ are SELTS such that $\mathbb{Q}_1 \approx_{se} \mathbb{Q}_2$ then $\mathbb{Q}_1 \models \neg(\Box \diamond \eta = tick) \vee F$ iff $\mathbb{Q}_2 \models \neg(\Box \diamond \eta = tick) \vee F$.*

Recalling from Section 2 that $\mathbb{Q} \approx_{se} \mathbb{Q}//\theta_w$, where $\mathbb{Q}//\theta_w$ is the weak state-event quotient system of $\mathbb{Q}$, Theorem 3 allows us to model check SESI formulas on a system's quotient system and infer the result for the original system. Additionally, Lemma 2 guarantees that our model reduction technique is compositionally consistent.

5 CONCLUSION

The main result of the paper is the development of a weak, compositionally consistent model reduction technique for a simple linear, discrete time temporal logic similar to RTTL [17]. The method is applicable to the subclass of State-Event Stuttering Invariant (SESI) formulas, which includes some formulas containing immediate operators.

The compositional consistency of the method is significant in that it allows one to avoid computing massive synchronous products before performing model reduction, by first doing model reduction on the component subsystems and *then* forming their synchronous product. Typically synchronous products grow as the product of the subsystem's state spaces so state reductions of subsystems have a multiplicative effect. In [19] this method's compositional consistency has been applied to a simple industrial real-time controller software verification problem, achieving close to an order of magnitude reduction in the time required to model check the system requirements.

References

1. A. Arnold. *Finite Transition Systems.* Prentice Hall, 1994.

2. S. Bensalem, A. Bouajjani, C.Loiseaux, and J. Sifakis. Property preserving simulations. In *Proc. of CAV'92*, LNCS 663, pages 260–275. Springer-Verlag, 1992.
3. B. Bolognesi and M. Caneve. Equivalence verification: Theory, algorithms and a tool. In *The Formal Description Technique LOTOS*, pages 303–326. North-Holland, 1989.
4. M.C. Brown, E.M. Clarke, and O. Grümberg. Characterizing kripke structures in temporal logic. In *TAPSOFT'87, vol. I*, LNCS 249, pages 256–270. Springer-Verlag, 1987.
5. J.R. Burch, E.M. Clarke, and K.L. McMillan. Symbolic model checking: 10^{20} states and beyond. *Information and Computation*, 98:142–170, 1992.
6. E.M. Clarke, E.A. Emerson, and A.P. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Trans. Prog. Lang. and Syst.*, 8(2):244–263, Apr 1986.
7. D. Dams, O. Grumberg, and R. Gerth. Abstract interpretation of reactive systems: Abstraction preserving $\forall$CTL*, $\exists$CTL* and CLT*. In E.-R. Olderog, editor, *Programming Concepts, Methods and Calculi*, pages 573–592. North-Holland, 1994.
8. E.A. Emerson, A.K. Mok, A.P. Sistla, and J. Srinivasan. Quantitative temporal reasoning. *Real-Time Systems*, 4:331–352, 1992.
9. S. Graf and C. Loiseaux. Property preserving abstraction under parallel composition. *TAPSOFT'93*, LNCS 668, pages 644–657. Springer-Verlag, 1993.
10. C.A.R. Hoare. *Communicating Sequential Processes.* International Series in Computer Science. Prentice-Hall International, Englewood Cliffs, NJ, 1985.
11. B. Josko. A context dependent equivalence relation between kripke structures. In *Proc. of 2nd Conf. on Computer Aided Verification*, LNCS 531, pages 204–213. Springer-Verlag, 1990.
12. R. Kaivola and A. Valmari. The weakest compositional semantic equivalence preserving nexttime-less linear temporal logic. In *Proc. of CONCOUR'92*, LNCS 630, pages 207–221. Springer-Verlag, 1992.
13. M. Lawford and W.M. Wonham. Equivalence preserving transformations of timed transition models. *IEEE Trans. Autom. Control*, 40:1167–1179, July 1995.
14. M. Lawford, W.M. Wonham, and J.S. Ostroff. State-event observers for labeled transition systems. In *Proc. of 33rd Conf. Decision and Control*, pages 3642–3648. Dec. 1994.
15. Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems.* Springer-Verlag, New York, 1992.
16. R. Milner. *Communication and Concurrency.* Prentice Hall, New York, 1989.
17. J.S. Ostroff. *Temporal Logic for Real-Time Systems.* RSP / Wiley, Taunton, UK, 1989.
18. J.S. Ostroff. Deciding properties of timed transition models. *IEEE Trans. Parallel and Distributed Systems*, 1(2):170–183, April 1990.
19. J.S. Ostroff. Abstraction and composition of discrete real-time systems. In *Proc. of CASE'95*, pages 370–380, 1995.
20. P.J. Ramadge and W.M. Wonham. Supervisory control of a class of discrete event processes. *SIAM J. Control Optim.*, 25(1):206–230, January 1987.
21. Y. Wang. *CCS + Time = an Interleaving Model for Real Time Systems*, LNCS 510, pages 217–228. Springer–Verlag, 1991.