STATE-EVENT OBSERVERS FOR LABELED TRANSITION SYSTEMS

M. Lawford and W.M. Wonham

Systems Control Group
Dept. of Electrical & Computer Eng.
University of Toronto
Toronto, ON M5S 1A4

J.S. Ostroff

Dept. of Computer Science
York University
4700 Keele Street
North York, ON M3J 1P3

ABSTRACT

In many Discrete-Event Systems (DES) both state and event information are of importance to the systems designer. As a first step towards obtaining hierarchical models of systems, the behavior of Discrete-Event Systems with unobservable transitions and state output maps is considered. Observers for deterministic DES are generalized to nondeterministic DES and characterized using the join semilattice of compatible partitions of a transition system. This characterization points to efficient algorithms for computing both strong and weak state-event observers as solutions to the Relational Coarsest Partition problem (RCP). The strong and weak observation equivalences of Milner are shown to be special cases of our observers under the trivial (constant) state output map.

1. Introduction

Research on discrete-event systems (DES) has led to renewed appreciation of control architecture - decentralized and hierarchical decomposition - for the effective modeling of large systems. In theoretical treatment, such architectural features are brought in through standard algebraic constructs, namely unions, products and quotient structures of the state sets involved. Inasmuch as architecture amounts to decomposition of information transfer and decision making, the systemic notions of observation and observer are fundamental. These find their algebraic setting in lattices of equivalence relations (partitions), and the associated sublattices of congruences with respect to the dynamic flow. Thus in approaching any new class of state transition structures, a first item of business is to clarify the algebraic structure of observers (congruences) along with their computational complexity. Because, in general, equivalence is undecidable, these issues tend to be both nontrivial and of practical interest.

In this paper we generalize previous observers - well known (under various guises) in either the control or process algebra literature - to a unified construct that we call a state-event observer. In this treatment both state changes and output events (or event signals) are assigned equal status, thus allowing a flexible modelling approach to DES in which both state- and event-based control are equally natural.

We recall the duality of states and events. Event-based models include process-algebraic theory such as those from [1], as well as control-theoretic approaches such as [2], [3]. States are really only viewed as a way of keeping track of what sequences of events have been executed and what future events are possible. Quotient structure is induced by projection of languages. On the other hand in [4],[5], state structure is preeminent, and behavior treated as sequences of states or groups of states. For instance state charts [5],[6] offer a visual representation (nested boxes and arrows) of state set decomposition via nested products and disjoint unions, in principle to arbitrary depth. Of course the transition structure and control must admit compatible decomposition for the method to be computationally attractive, and to admit quotient structures induced by suitable state-transition homomorphisms.

In many applications both state occupancy and event sequencing are important, and so we need quotients with respect to both. One instance is Timed Transition Models (TTMs) [7], which express behavior such as: "Do α only when $y = 2$ for 3 or more '*tick*s' of the clock." In [8] the authors adapted to TTMs the event-based observation equivalence of [1] by projecting TTM states (the state assignments of [7]) to their factors defined by selected subsets of data variables. Observable events are just those TTM state changes that affect the variables in question, and the event labels themselves are "projected out". The class of projections for which a quotient can be defined was severely restricted; but we shall show how this situation can be improved on.

In this paper we introduce so-called strong and weak state-event observers for Labeled Transition Systems (LTS) (the underlying model of many DES formalisms including TTMs); state output maps and event projections play symmetric roles. Our observers (congruences) induce consistent high-level abstractions (quotients) so that, just as in [2], control designed at the abstract level can be consistently implemented at the detailed ('real-world') level. The development of strong observers and their quotient systems parallels the results on indistinguishability of LTS in [9]. On the basis of [10],[11],[12] we are able to appeal to efficient polynomial-time algorithms for computing our observers on finite-state LTS.

2. Preliminaries

In this section Labeled Transition Systems (LTS) will be used as our model of a Discrete Event System (DES). The lattice of congruences of a deterministic transition system and its role in characterizing the (strong) state observers of [4] are reviewed.

2.1 Labeled Transition Systems

LTS can be used to compare different notions of equivalence proposed for concurrent systems, including TTMs.

Definition 2.1 *A* **Labeled Transition System** *is a 4-tuple* $\mathbb{Q} = (Q, \Sigma, R_\Sigma, q_0)$ *where* Q *is an at most countable set of states,* Σ *is an at most countable set of elementary actions or events,* $R_\Sigma = \{\xrightarrow{\alpha} : \alpha \in \Sigma\}$ *is a set of binary relations on* Q*, and* $q_0 \in Q$ *is the initial state.*

In the above definition if $\alpha \in \Sigma$ and $q, q' \in Q$, then $q \xrightarrow{\alpha} q'$ means that the LTS 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 LTS $\mathbb{Q}$. When the LTS 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.

2.2 State Observers for Deterministic LTS

In [4] the author considers LTS of the form $\mathbb{Q} = (Q, \{\alpha\}, \{\xrightarrow{\alpha}\}, q_0)$, where $\xrightarrow{\alpha}$, the lone transition relation, can be represented as a function $\alpha : Q \to Q$. In this case the LTS is viewed as a discrete time dynamical system, given by $x(0) = q_0$ and $x(t+1) = \alpha(x(t))$, where it is the sequence of states generated by the LTS that is of interest. An output map with no special structure, $P_Q : Q \to R$, is given. Then two states $q, q' \in Q$ produce the same output observation precisely when $P_Q(q) = P_Q(q')$.

Denote the set of all equivalence relations on Q by $Eq(Q)$. Any state output map $P_Q : Q \to R$ induces an equivalence relation $\ker(P_Q) \in Eq(Q)$, the equivalence kernel of P_Q, given by

$$(q_1, q_2) \in \ker(P_Q) \text{ if and only if } P_Q(q_1) = P_Q(q_2)$$

Similarly, any $\theta \in Eq(Q)$ defines a canonical output map $\theta : Q \to Q/\theta$, which projects each $q \in Q$ onto its θ-cell (equivalence class). $Eq(Q)$ 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.

Given a deterministic LTS $\mathbb{Q}$ as defined above, $\theta \in Eq(Q)$ is a *congruence* of $\mathbb{Q}$ iff $(q, q') \in \theta$ implies $(\alpha(q), \alpha(q')) \in \theta$. We let $Con(\mathbb{Q})$ denote the set of all congruences of $\mathbb{Q}$. As noted in [4], $Con(\mathbb{Q})$ forms a complete sublattice of $Eq(Q)$. Thus $Con(\mathbb{Q})$ is closed under $\wedge$ and $\vee$, and given any $\mathcal{F} \subseteq Con(\mathbb{Q})$, $\sup(\mathcal{F})$ exists as an element of $Con(\mathbb{Q})$.

Definition 2.2 *Given a deterministic LTS* $\mathbb{Q}$ *as defined above and a state output map* $P_Q : Q \to R$*, the* strong state observer*,* $\omega(\mathbb{Q}, P_Q)$*, is defined to be*

$$\omega(\mathbb{Q}, P_Q) = \sup\{\theta \in Con(\mathbb{Q}) : \theta \leq \ker(P_Q)\}$$

When $\mathbb{Q}$ and P_Q are clear from the context we will simply write ω for $\omega(\mathbb{Q}, P_Q)$. The existence and uniqueness of ω are an immediate result of $Con(\mathbb{Q})$ being a complete sublattice of $Eq(Q)$. Here ω is the coarsest congruence that is finer than the equivalence kernel of P_Q. For $(q, q') \in \omega$, $\omega \leq \ker(P_Q)$ implies $P_Q(q) = P_Q(q')$ while $\omega \in Con(\mathbb{Q})$ so $(\alpha(q), \alpha(q')) \in \omega$ and hence $P_Q(\alpha(q)) = P_Q(\alpha(q'))$. Thus if $(q, q') \in \omega$, then q and q' produce the same current state output and sequence of future state outputs.

From an informational standpoint, ω represents the minimum information you need about the current state of the system to be able to predict the future state outputs.

3. Strong State-Event Observers

We now wish to generalize the observers for deterministic LTS with a single transition function to observers for general LTS with multiple nondeterministic transition relations. In this case 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 $P_Q : Q \to R$.

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

Later τ will be used to denote unobservable events but for now we assume that all τ transitions are observable. In this case 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.

Congruences are defined only for transition functions but we are now dealing with nondeterministic transition relations so we must find a class of partitions that plays the role of congruences for nondeterministic relations.

Definition 3.1 *(cf.[12] Given a LTS* $\mathbb{Q} = (Q, \Sigma, R_\Sigma, q_0)$*, 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*,*

$$\alpha(q) \cap C_j \neq \emptyset \textit{ iff } \alpha(q') \cap C_j \neq \emptyset$$

The set of all compatible partitions for the LTS $\mathbb{Q}$ *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 $(q_1, q_1') \in \theta$. The reader familiar with Milner's observation equivalence will note that compatible partitions are special cases of bisimulation relations and have been used for the efficient computation of (event) observation equivalence of LTS [10], [12]. We will have more to say about this later. First we will see if $CP(\mathbb{Q})$ has any special algebraic structure.

In the case of congruences, $Con(\mathbb{Q})$ forms a complete sublattice of $Eq(Q)$ so perhaps we can expect something similar for $CP(\mathbb{Q})$. Consider Figure 1. It is easy to verify that θ_1, θ_2 and $\theta_1 \vee \theta_2$ are compatible partitions of the given LTS but $\theta_1 \wedge \theta_2$ is not. Thus $CP(\mathbb{Q})$ is not closed under the $\wedge$ operation of $Eq(Q)$. The following

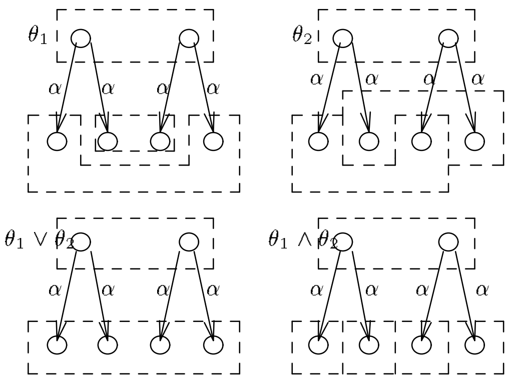


Figure 1: $CP(\cdot)$ is closed under $\vee$ but not $\wedge$

Lemma claims that $CP(\mathbb{Q})$ is closed under the $\vee$ operator of $Eq(Q)$ so although $CP(\mathbb{Q})$ is not a complete sublattice of $Eq(Q)$, it does retain the complete join semilattice property of $Con(\mathbb{Q})$ that was used in defining state observers in the previous section. We were led to expect a join semilattice structure for defining observers on systems with nondeterministic transition relations from Wong's investigation of the algebraic properties of hierarchy in [3].

Lemma 3.1 *For a given LTS $\mathbb{Q} = (Q, \Sigma, R_\Sigma, q_0)$, the set of compatible partitions for $\mathbb{Q}$, $CP(\mathbb{Q})$, forms a complete sub-semilattice (with respect to join) of $Eq(Q)$.*

An immediate result of Lemma 3.1 is that for any $\mathcal{F} \subseteq CP(\mathbb{Q})$, there is a unique supremal element $\omega := \sup(\mathcal{F})$ and $\omega \in CP(\mathbb{Q})$. We are now in a position to characterize a strong state-event observer for any given LTS and output map.

Definition 3.2 *Given a LTS $\mathbb{Q} = (Q, \Sigma, R_\Sigma, q_0)$ and a state output map $P_Q : Q \to R$ (with no special structure), the* strong state-event observer, *$\omega_s(\mathbb{Q}, P_Q)$ is defined to be*

$$\omega_s(\mathbb{Q}, P_Q) = \sup\{\theta \in CP(\mathbb{Q}) : \theta \le \ker(P_Q)\}$$

When $\mathbb{Q}$ and P_Q are clear from the context we will simply write ω_s for $\omega_s(\mathbb{Q}, P_Q)$. Similar to the case of the state observers of Section 2, ω_s is the coarsest compatible partition of $\mathbb{Q}$ that is finer than the equivalence kernel of P_Q. Thus for $(q, q') \in \omega_s$ we have $P_Q(q) = P_Q(q')$ so q and q' produce the same current state output. Now suppose that $q \xrightarrow{\alpha} q_1$, thereby producing event output α and state output $P_Q(q_1)$. Since $\omega_s \in CP(\mathbb{Q})$ there exists $q_1' \in \alpha(q')$ such that $(q_1, q_1') \in \omega_s$. Hence $q' \xrightarrow{\alpha} q_1'$ and $P_Q(q_1) = P_Q(q_1')$ so q' can generate identical state and event outputs to q. As was the case with state observers, ω_s represents the minimum information one needs about the current state to be able to predict all possible future state and event outputs.

The Relational Coarsest Partition problem (RCP) (as stated in [10]) can be phrased as "Given a LTS $\mathbb{Q} = (Q, \Sigma, R_\Sigma, q_0)$ and θ_0, an initial partition of Q, find the coarsest compatible partition of $\mathbb{Q}$ that is finer than θ_0 (ie. find $\sup\{\theta \in CP(\mathbb{Q}) : \theta \le \theta_0\}$). Thus ω_s is the solution to the RCP with $\theta_0 := \ker(P_Q)$. In the special case when $\theta_0 = \ker(P_Q) = \nabla$ (no state information is provided by the state output map), the solution of the RCP is Milner's strong observation equivalence $\sim$ [10]. Therefore when there are only event outputs and no state outputs, our strong state-event observers reduce to Milner's strong observation equivalence.

An $O(m \log n)$ algorithm, where m is the size of R_Σ (the number of related pairs) and $n = |Q|$, for computing $\sim$ for finite state LTS, based upon Paige and Tarjan's solution to the (mono)-RCP (RCP with only one relation present) [11], can be found in [12]. In this case, θ_0 is of course ∇. This algorithm is easily adapted to computing ω_s without any change in complexity (assuming $\ker(P_Q)$ is provided) by allowing the initial partition for the RCP to be $\ker(P_Q)$ which, in general, is not ∇. This close connection with $\sim$ leads us to write $q \sim_{P_Q} q'$ when$(q, q') \in \omega_s$ and say that q is strong state-event observation equivalent to q'.

What differentiates our work from that of [10] and [12], is the use, as suggested in [9], of a nontrivial initial partition in the RCP, to consider both event and state outputs. The consideration of both state and event outputs takes on even greater significance when we consider weak state-event observers in the next section. With little additional effort we can adapt [10] and [12] to provide an efficient algorithm for computing weak state-event observers.

As a generalization of congruences, we might expect that compatible partitions can be used to construct quotient systems of nondeterministic LTS.

Definition 3.3 *Given a LTS $\mathbb{Q} := (Q, \Sigma, R_\Sigma, q_0)$, for $\theta \in CP(\mathbb{Q})$, we define the* quotient system *of $\mathbb{Q}$ by θ, $\mathbb{Q}/\theta$, as follows:*

$$\mathbb{Q}/\theta := (Q/\theta, \Sigma, R_\Sigma/\theta, q_0/\theta)$$

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 $\alpha^{\mathbb{Q}/\theta}(q/\theta) = \alpha^{\mathbb{Q}}(q)/\theta = \{q_1/\theta \in Q/\theta : q_1 \in \alpha^{\mathbb{Q}}(q)\}$.

The remainder of this section is dedicated to proving that the quotient system generated by the compatible partition ω_s is the "unique" minimal state LTS that has a state output map that makes it strongly state-event (observationally) equivalent to the original system. To do this we first have to have a definition of when two LTS, with associated state output maps, are state-event equivalent.

As was the case with observation equivalence in [12], strong state-event observation equivalence can be extended to a relation $\sim_{se}$ between two LTS having disjoint state sets and state output maps. This is done by forming the union of the transition systems and the union of the original systems' state output maps. The two LTS are then strongly state-event equivalent iff their initial states are strongly state-event observationally equivalent in the union system. More formally,

Definition 3.4 *Given two disjoint LTS $\mathbb{Q}_1 = (Q_1, \Sigma, R_\Sigma^1, q_{10})$ and $\mathbb{Q}_2 = (Q_2, \Sigma, R_\Sigma^2, q_{20})$ with state output maps $P_{Q_1} : Q_1 \to R$ and $P_{Q_2} : Q_2 \to R$, we define the union of $\mathbb{Q}_1$ and $\mathbb{Q}_2$ to be*

$$\mathbb{Q}_1 \cup \mathbb{Q}_2 := (Q_1 \cup Q_2, \Sigma, R_\Sigma^1 \cup R_\Sigma^2, q_{10})$$

while the union of the state output functions, $P_{Q_1} \cup P_{Q_2} : Q_1 \cup Q_2 \to R$, is given by $(P_{Q_1} \cup P_{Q_2})|Q_i = P_{Q_i}$. Then

$\mathbb{Q}_1$ *under state output map* P_{Q_1} *is* strongly state-event equivalent *to* $\mathbb{Q}_2$ *under state output map* P_{Q_2}, *written* $(\mathbb{Q}_1, P_{Q_1}) \sim_{se} (\mathbb{Q}_2, P_{Q_2})$, *iff* $(q_{10}, q_{20}) \in \omega_s(\mathbb{Q}_1 \cup \mathbb{Q}_2, P_{Q_1} \cup P_{Q_2})$.

In the definition of $\mathbb{Q}_1 \cup \mathbb{Q}_2$ we have made the arbitrary choice of q_{10} as the initial state. Either initial state will do for proving properties of quotient systems.

The notion of a homomorphism of a LTS will, of course, play a central role in obtaining our results about quotient systems. The nondeterministic transition relations lead us to extend the notion of homomorphism in much the same way that we extended congruences of deterministic LTS to compatible partitions of nondeterministic LTS. A LTS homomorphism requires that any α move in the low level system can be matched by an α move in the high level system and vice versa. It also requires that the initial state of the low level LTS be mapped to the initial state of the high level LTS. In the definition of a LTS homomorphism we use the fact that any function $h : Q_1 \to Q_2$ induces a function at the power set level, $h_* : \mathcal{P}(Q_1) \to \mathcal{P}(Q_2)$. For $Q \subseteq Q_1$, $h_*(Q) := \{h(q) : q \in Q\}$.

Definition 3.5 *Given two LTS* $\mathbb{Q}_1 = (Q_1, \Sigma, R^1_\Sigma, q_{10})$ *and* $\mathbb{Q}_2 = (Q_2, \Sigma, R^2_\Sigma, q_{20})$, *a mapping* $h : Q_1 \to Q_2$ *is a* LTS homomorphism *from* $\mathbb{Q}_1$ *to* $\mathbb{Q}_2$ *if*

(i) $h(q_{10}) = q_{20}$

(ii) For all $\alpha \in \Sigma, q_1 \in Q_1$ $h_*(\alpha^{\mathbb{Q}_1}(q_1)) = \alpha^{\mathbb{Q}_2}(h(q_1))$

In this case we will write $h : \mathbb{Q}_1 \to \mathbb{Q}_2$. *Henceforth homomorphism will be understood to mean LTS homomorphism.*

When state output maps $P_{Q_1} : Q_1 \to R$ *and* $P_{Q_2} : Q_2 \to R$ *are associated with* $\mathbb{Q}_1$ *and* $\mathbb{Q}_2$ *respectively, we say that* $h : \mathbb{Q}_1 \to \mathbb{Q}_2$ *is an* output compatible homomorphism *or OC homomorphism if for all* $q_1 \in Q_1$, $P_{Q_1}(q_1) = P_{Q_2} \circ h(q_1)$ *We will use the notation* $h : (\mathbb{Q}_1, P_{Q_1}) \to (\mathbb{Q}_2, P_{Q_2})$ *to emphasize the role of the state output maps.*

If $\mathbb{Q}_2$ is an output compatible homomorphic image of $\mathbb{Q}_1$, any event and associated state output change in $\mathbb{Q}_1$ can be matched in $\mathbb{Q}_2$. This situation leads us to expect that homomorphisms and compatible partitions are closely related.

Any compatible partition defines a homomorphism. For an LTS $\mathbb{Q} := (Q, \Sigma, R_\Sigma, q_0)$, any $\theta \in CP(\mathbb{Q})$ defines a (natural) homomorphism $\theta : \mathbb{Q} \to \mathbb{Q}/\theta$. Thus for state output map $P_Q : Q \to R$, if $\theta \in \{\theta \in CP(\mathbb{Q}) : \theta \leq \ker(P_Q)\}$ then $\theta : (\mathbb{Q}, P_Q) \to (\mathbb{Q}/\theta, P_{Q/\theta})$ is an OC homomorphism and $P_{Q/\theta} : Q/\theta \to R$ is the unique state output map such that $P_Q = P_{Q/\theta} \circ \theta$.

Similarly, there is a compatible partition associated with every homomorphism. It can be shown that if $h : \mathbb{Q}_1 \to \mathbb{Q}_2$ is a homomorphism then $\ker(h) \in CP(\mathbb{Q})$. We can now talk about *output compatible partitions* – those partitions of a LTS that correspond to the kernel of an OC homomorphism of the LTS $\mathbb{Q}$ for a given state output map P_Q.

We are now ready to give the main result of this section, which states that two LTS, with their respective state output maps, are strongly state-event equivalent iff they share an output compatible homomorphic image.

Theorem 3.1 *For two disjoint LTS* $\mathbb{Q}_1$ *and* $\mathbb{Q}_2$ *as in Definition 3.5, with state output maps* $P_{Q_1} : Q_1 \to R$ *and* $P_{Q_2} : Q_2 \to R$, *we have* $(\mathbb{Q}_1, P_{Q_1}) \sim_{se} (\mathbb{Q}_2, P_{Q_2})$ *iff there exists a LTS* $\mathbb{Q}_3$ *with state output map* $P_{Q_3} : Q_3 \to R$ *for which there are OC homomorphisms* $h_1 : (\mathbb{Q}_1, P_{Q_1}) \to (\mathbb{Q}_3, P_{Q_3})$ *and* $h_2 : (\mathbb{Q}_2, P_{Q_2}) \to (\mathbb{Q}_3, P_{Q_3})$.

Employing a method similar to that used in [9], as a corollary to this theorem, we obtain the result that when $\mathbb{Q}$ is reachable, $\mathbb{Q}/\omega_s$ is the unique (up to isomorphism) minimal state LTS for which there exists a state output map $P_{\mathbb{Q}/\omega_s}$ such that $(\mathbb{Q}, P_Q) \sim_{se} (\mathbb{Q}/\omega_s, P_{Q/\omega_s})$. It can then be shown that two reachable LTS with state output maps are strongly state-event equivalent iff their strong state-event observer quotient systems are isomorphic.

4. Weak State-Event Observers

Often in Discrete Event Systems it is the case that 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 will see how it is applied in the event setting of Milner's weak observation equivalence. Again we will see that (event) observation equivalence becomes the special case of our setting in which $\ker(P_Q) = \nabla$.

Consider a LTS $\mathbb{Q} := (Q, \Sigma, R_\Sigma, q_0)$. In the style of [12], 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,

$$\mathbb{Q}' := (Q, \Sigma, R'_\Sigma, q_0)$$

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 $\tau^{\mathbb{Q}'}(q) = \{q_1 \in Q : q \overset{\epsilon}{\Rightarrow} q_1 \text{ in } \mathbb{Q}\}$.

In [10], two states are shown to be weakly observation equivalent in $\mathbb{Q}$ in the sense of [1], written $q \approx q'$, iff the states are strongly observation equivalent ($q \sim q'$) in $\mathbb{Q}'$. Thus we have $\approx := \sup(CP(\mathbb{Q}'))$. In this case $\approx$ 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 LTS $\mathbb{Q}$ and a state output map $P_Q : Q \to R$, we 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 $P_Q(q) = P_Q(q')$ then there is no noticeable change in the system output. If, on the other hand, $q \xrightarrow{\tau} q'$ and $P_Q(q) \neq P_Q(q')$ 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 a given LTS $\mathbb{Q}$ and state output map P_Q, an unobservable move from q to q', written $q \Rightarrow_{P_Q} q'$ 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 $j = 0, 1, \ldots, n$ $P_Q(q_j) = P_Q(q) = P_Q(q')$

Thus the relation $\Rightarrow_{P_Q}$ is the transitive closure of the τ relation within each cell of $\ker(P)$. By convention $q \Rightarrow_{P_Q} q$ always holds. While the $\Rightarrow_{P_Q}$ relation captures a relation which is indistinguishable from the case when $q \xrightarrow{\tau} q'$ and $P_Q(q) = P_Q(q')$, we now wish to define a relation which captures both this case and the case when $q \xrightarrow{\tau} q'$ and $P_Q(q) \neq P_Q(q')$. We say that q moves to q' without an event output, written $q \overset{\epsilon}{\Rightarrow}_{P_Q} q'$, iff $q = q'$, or there exist $q_1, q_2 \in Q$ such that

$$q \Rightarrow_{P_Q} q_1 \xrightarrow{\tau} q_2 \Rightarrow_{P_Q} q'$$

By definition $q \overset{\epsilon}{\Rightarrow}_{P_Q} q$. The relation $\overset{\epsilon}{\Rightarrow}_{P_Q}$ is the transitive closure of $\xrightarrow{\tau}$ subject to the restriction that at most one boundary of the partition $\ker(P_Q)$ is crossed. If $q \overset{\epsilon}{\Rightarrow}_{P_Q} q'$, then no output events are generated and there is at most one change in the state output.

We now define a relation similar to $\overset{\epsilon}{\Rightarrow}_{P_Q}$ 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 $q \overset{\alpha}{\Rightarrow}_{P_Q} q'$ iff there exist $q_1, q_2 \in Q$ such that

$$q \Rightarrow_{P_Q} q_1 \xrightarrow{\alpha} q_2 \Rightarrow_{P_Q} q'$$

Thus if $q \overset{\alpha}{\Rightarrow}_{P_Q} q'$, then q moves within a cell of $\ker(P_Q)$ via unobservable τ transitions, then performs an α transition which could possibly (but not necessarily) take us to a new cell of $\ker(P_Q)$ and then the system again moves unobservably via τ transitions within the current cell. We emphasize that if a boundary of $\ker(P_Q)$ is crossed when $q \overset{\alpha}{\Rightarrow}_{P_Q} q'$, then it is only crossed by the α transition.

There are four different types of one step moves that a LTS $\mathbb{Q}$ with output map P_Q can make and each of these moves can be matched by a double arrow relation defined above. In the following let q and q' be elements of Q such that $P_Q(q) = P_Q(q')$. Then the system can:

1. Make an unobservable transition within a cell of $\ker(P_Q)$ ($q \xrightarrow{\tau} q_1$ and $P_Q(q) = P_Q(q_1)$). State q' can make the move $q' \overset{\epsilon}{\Rightarrow}_{P_Q} q_1'$ with $P_Q(q_1') = P_Q(q_1)$.

2. Make a τ transition that moves from one cell of $\ker(P_Q)$ to another ($q \xrightarrow{\tau} q_1$ and $P_Q(q) \neq P_Q(q_1)$). State q' can make the move $q' \overset{\epsilon}{\Rightarrow}_{P_Q} q_1'$ with $P_Q(q_1') = P_Q(q_1)$.

3. Make an observable transition α within a cell of $\ker(P_Q)$ ($q \xrightarrow{\alpha} q_1$ and $P_Q(q) = P_Q(q_1)$). State q' can make the move $q' \overset{\alpha}{\Rightarrow}_{P_Q} q_1'$ with $P_Q(q_1') = P_Q(q_1)$.

4. Make an observable transition α that moves from one cell of $\ker(P_Q)$ to another ($q \xrightarrow{\alpha} q_1$ and $P_Q(q) \neq P_Q(q_1)$). State q' can make the move $q' \overset{\alpha}{\Rightarrow}_{P_Q} q_1'$ with $P_Q(q_1') = P_Q(q_1)$.

Consider the state event sequences (1) of Section 3 from the point of view that only output (observable) events and changes in the state output are important. The first two sequences are indistinguishable when viewed from state and event outputs. In both sequences the event α and the state output change from r_1 to r_2 occur simultaneously. Hence $q_{11} \overset{\alpha}{\Rightarrow}_{P_Q} q_{13}$ and $q_{21} \overset{\alpha}{\Rightarrow}_{P_Q} q_{23}$ 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. In terms of our newly defined relations $q_{31} \overset{\epsilon}{\Rightarrow}_{P_Q} q_{32} \overset{\alpha}{\Rightarrow}_{P_Q} q_{33}$ but not $q_{31} \overset{\alpha}{\Rightarrow}_{P_Q} q_{33}$ 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 be able to distinguish the first two sequences from the third. Assume that r_2 is a bad state output that we wish to avoid and that α is a controllable event that can be disabled as in [13]. 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 4.1 *Given a LTS $\mathbb{Q} = (Q, \Sigma, R_\Sigma, q_0)$ and a state output map $P_Q : Q \to R$, the* weak state-event observer, *$\omega_w(\mathbb{Q}, P_Q)$ is defined to be*

$$\omega_w(\mathbb{Q}, P_Q) = \sup\{\theta \in CP(\mathbb{Q}'_{P_Q}) : \theta \leq \ker(P_Q)\}$$

Here

$$\mathbb{Q}'_{P_Q} := (Q, \Sigma, R'_\Sigma, q_0)$$

where R'_Σ is defined as follows. For all $\alpha \in \Sigma_o$, $q \xrightarrow{\alpha} q'$ in $\mathbb{Q}'_{P_Q}$ iff $q \overset{\alpha}{\Rightarrow}_{P_Q} q'$ in $(\mathbb{Q}, P_Q)$ and $q \xrightarrow{\tau} q'$ in $\mathbb{Q}'_{P_Q}$ iff $q \overset{\epsilon}{\Rightarrow}_{P_Q} q'$ in $(\mathbb{Q}, P_Q)$.

By Lemma 3.1 ω_w always exists and is unique. Note that in $\mathbb{Q}'_{P_Q}$ the transition relations are dependent upon P_Q so ω_w is not just Milner's observation equivalence with a different initial partition (as was the case for strong state-event observers). It is easy to see that in the case when $\ker(P_Q) = \nabla$ then ω_w is in fact $\approx$, Milner's weak observation equivalence, since $\mathbb{Q}'_{P_Q}$ becomes $\mathbb{Q}'$. As was the case for strong state-event equivalence, when $(q, q') \in \omega_w$ for a given $\mathbb{Q}$ and P_Q, we will write $q \approx_{P_Q} q'$, read "q is weak state-event observation equivalent to q'". The $O(n^3)$ algorithm ($n = |Q|$) for computing Milner's weak observation equivalence of finite state LTS given in [12] can be easily adapted to provide an $O(n^3)$ algorithm for ω_w. After computing $\mathbb{Q}'_{P_Q}$, the $O(m \log n)$ RCP algorithm of [11] can be employed to compute ω_w giving an overall complexity of $O(n^3)$.

Similar to the case of the state observers of Section 2 and the strong state-event observers of Section 3, ω_w is the coarsest compatible partition of $\mathbb{Q}'_{P_Q}$ that is finer than the equivalence kernel of P_Q. Although the double arrow relations used to construct $\mathbb{Q}'_{P_Q}$ may or may not cross a boundary of the partition of $\ker(P_Q)$, the use of $\ker(P_Q)$ as the initial partition detects when a change in state output occurs. Thus for $(q, q') \in \omega_w$ we have $P_Q(q) = P_Q(q')$ so q and q' produce the same current state output. Now suppose that $q \xrightarrow{\alpha} q_1$ in $\mathbb{Q}$, thereby producing event output α and state output $P_Q(q_1)$. Then $q \overset{\alpha}{\Rightarrow}_{P_Q} q_1$ in $\mathbb{Q}$ so $q \xrightarrow{\alpha} q_1$ in $\mathbb{Q}'_{P_Q}$, and since $\omega_w \in CP(\mathbb{Q}'_{P_Q})$ there exists $q_1' \in \alpha^{\mathbb{Q}'_{P_Q}}(q')$ such that $(q_1, q_1') \in \omega_w$. Hence $q' \xrightarrow{\alpha} q_1'$ in $\mathbb{Q}'_{P_Q}$ and $P_Q(q_1) = P_Q(q_1')$. But then in $\mathbb{Q}$, $q' \overset{\alpha}{\Rightarrow}_{P_Q} q_1'$. Thus q' can generate state and event outputs that are indistinguishable from those produced from q. As was the case with strong state-event observers, ω_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 LTS $\mathbb{Q}$ with state output P_Q is just the strong state-event observer for the pair $(\mathbb{Q}'_{P_Q}, P_Q)$, we can use the results of the previous section to derive similar results about what we will term *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 $q \overset{\alpha}{\Rightarrow}_{P_Q} q'$ are indistinguishable.

Definition 4.2 *Given an LTS* $\mathbb{Q} := (Q, \Sigma, R_\Sigma, q_0)$ *with state output map* $P_Q : Q \to R$, *for* $\theta \in CP(\mathbb{Q}'_{P_Q})$, $\mathbb{Q}//\theta := \mathbb{Q}'_{P_Q}/\theta$ *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 LTS by forming the union of disjoint LTS (see Definition 3.4).

Lemma 4.1 $(\mathbb{Q}_1, P_{Q_1}) \sim_{se} (\mathbb{Q}_2, P_{Q_2})$ *implies* $(\mathbb{Q}_1, P_{Q_1}) \approx_{se} (\mathbb{Q}_2, P_{Q_2})$.

We can now obtain the main result of this section.

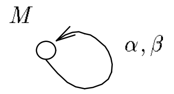
Theorem 4.1 *For any reachable LTS and state output map* $(\mathbb{Q}, P_Q)$, *the weak quotient system* $\mathbb{Q}//\omega_w$ *is a minimal state LTS for which there exists a state output map* $P_{\mathbb{Q}/\omega_w}$ *such that* $(\mathbb{Q}, P_Q) \approx_{se} (\mathbb{Q}//\omega_w, P_{Q/\omega_w})$.

The proof follows from Theorem 3.1, Lemma 4.1 and the idempotence of the $'_{P_Q}$ operator.

In general $\mathbb{Q}//\omega_w$ is one of many possible minimal state LTS that can be equivalent to $(\mathbb{Q}, P_Q)$ but that differ in the definition of their transition relations. Uniqueness of a minimal state equivalent LTS is lost in the weak state-event observation equivalence setting because of the use of the many-to-one $'_{P_Q}$ operator in Definition 4.1.

5. Example

In this section we present a small example. The weak state-event observer theory will be applied to the Timed Transition model (TTM) M of Figure 2.



$$\alpha := (true, [z : z \oplus_2 1], 0, 1)$$
$$\beta := (z = 0, [y : y \oplus_3 1], 1, \infty)$$
$$\Theta := (y = z = 0)$$

Figure 2: Example TTM M

A TTM is a guarded transition system with lower and upper time bounds on the transitions that relate to the number of occurrences of the special transition *tick*. For M there are three transitions, α, β and *tick*, and two program variables, y and z. The initial condition Θ specifies that M starts with both y and z set to 0. Now consider the transition $\alpha := (true, [z : z \oplus_2 1], 0, 1)$. The guard or "enablement condition" of α is $true$, hence the transition is always enabled. When the transition α occurs, it has the effect z becomes $z \oplus_2 1$ (here $\oplus_n$ denotes addition mod n). The lower and upper time bounds for α are 0 and 1 respectively. For α to occur, its guard condition must evaluate to $true$ continuously for at least 0 *tick* transitions and if its guard remains $true$ after one *tick*, it will be forced to occur before the next *tick* event. Since α's guard transition always evaluates to $true$, the above time bounds force at least one, to at most an arbitrarily large finite number of α's to occur between successive *tick*s of the "clock". In the case of $\beta := (z = 0, [y : y \oplus_3 1], 1, \infty)$, the value of z must be 0 for at least one *tick* before β can occur. The upper time bound of ∞ indicates that even if β is continuously enabled for arbitrarily many occurrences of *tick*, it is never forced to occur. If β does occur then y changes to $y \oplus_3 1$.

The LTS representing the "trajectories" of M is shown in Figure 3. The reader is referred to [7] for complete details of the semantics of TTMs used to obtain the LTS. Beside each state of the LTS in Figure 3, we write the

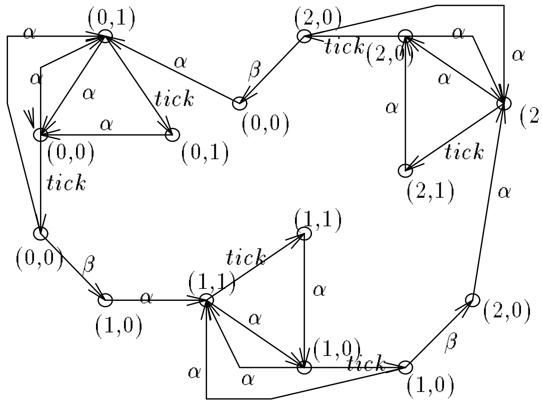


Figure 3: LTS generated by TTM M

ordered pair (y, z) to give the current value of the program variables y and z. The initial state of the LTS (q_0) is the state with the entering arrow.

Suppose we are interested in the timed behavior of M under the state output map,

$$P_Q(q) := \begin{cases} a, & y = 2 \\ b, & \text{otherwise} \end{cases}$$

The partition $\ker(P_Q)$ induced on M's LTS is shown in Figure 4. In this case the event *tick* remains observable while α and β are replaced in the LTS with unobservable τ transitions since it is only their effect on the state output that is of interest. Once the relations $\overset{\epsilon}{\Rightarrow}_{P_Q}$ and $\overset{tick}{\Rightarrow}_{P_Q}$ are determined, we can compute the weak state observer ω_w, the refinement of $\ker(P_Q)$ shown as dotted lines in Figure 4.

To understand how ω_w is obtained from $\ker(P_Q)$, consider the individual states of the LTS. States 9 and 14 are the only two states that are the sources of sequences of unobservable τ transitions that change the state output (eg. $9 \overset{\epsilon}{\Rightarrow}_{P_Q} 10$ and $10 \in P_Q^{-1}(a)$). Hence 9 and 14 are sectioned off from their respective cells of $\ker(P_Q)$. When the relation $\overset{tick}{\Rightarrow}_{P_Q}$ is considered, further refinements of $\ker(P_Q)$ result. State 4 can reach state 9 via silent τ transitions within a cell of $\ker(P_Q)$ and a *tick* (eg. $4 \overset{tick}{\Rightarrow}_{P_Q} 9$) while also being able to access states 1, 2, 3 and 15, states that cannot reach state 9 via the $\overset{tick}{\Rightarrow}_{P_Q}$ relation. As a result

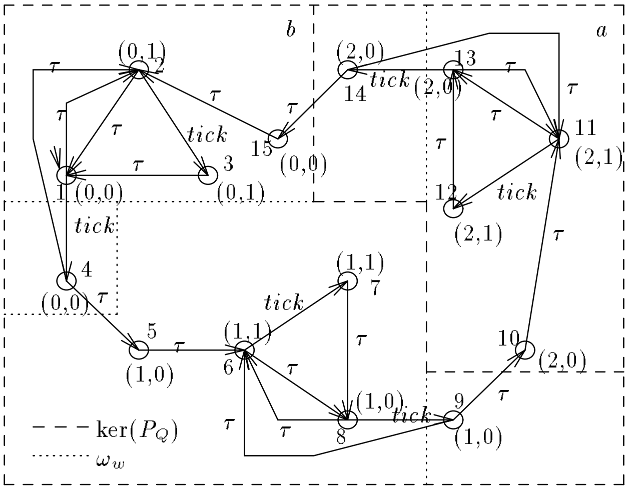


Figure 4: ker(P_Q) and resulting ω_w for LTS generated by TTM M

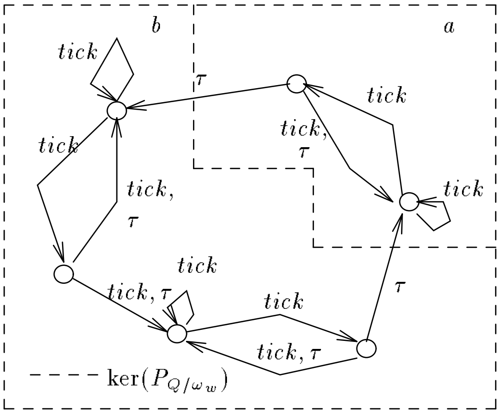


Figure 5: Weak Quotient system generated by ω_w

4 is split off from the other states of ker(P_Q). The rest of the refinement of ker(P_Q) proceeds in a similar fashion. It is left to the reader to verify that the partition ω_w shown in Figure 4 is indeed a compatible partition for the relations $\overset{\epsilon}{\Rightarrow}_{P_Q}$ and $\overset{tick}{\Rightarrow}_{P_Q}$ as defined in the previous section.

Figure 5 presents the weak quotient system with respect to the weak state-event observer ω_w of the LTS for M under state output map P_Q.

6. Conclusion

The general state-event setting of LTS with state output maps and unobservable transitions is considered as a way of hiding complexity and providing hierarchy in the sense of quotient systems. This setting leads to the development of state-event observers that are applicable to a wide variety of problems since LTS are the underlying model of many discrete event formalisms.

State-event observers of LTS represent a unifying framework for observers, and thereby hierarchy, in state and event based settings, enabling us to define observers in DES settings where both states and events are important (eg. Ostroff's TTMs). This unification of state and event methods is evidenced by the fact that the state observers of [4] and event based observation equivalences of Milner [1] are both special cases of state-event observers. The unification of methodologies is obtained through the algebraic characterization of strong and weak state-event observers using the upper semilattice of compatible partitions of a LTS. The algebraic characterization then enables appeal to efficient algorithms for computing state-event observers based upon the Relational Coarsest Partition problem.

7. References

[1] R. Milner, *A Calculus of Communicating Systems*, vol. 92 of *LNCS*, Springer-Verlag, New York, 1980.

[2] H. Zhong and W.M. Wonham, "On the consistency of hierarchical supervision in discrete-event systems", *IEEE Trans. Autom. Control*, vol. 35, pp. 1125–1134, Oct. 1990.

[3] K.C. Wong, *Control Architecture of Discrete-Event Systems: An Algebraic Approach*, PhD thesis, Dept. of El. Eng., Univ. of Toronto, Canada, June 1994.

[4] W.M. Wonham, "Towards an abstract internal model principle", *IEEE Trans. Systems Man and Cybernetics*, vol. 6, pp. 730–752, Nov. 1976.

[5] D. Harel, "Statecharts: A visual formalism for complex systems", *Science of Computer Programming*, vol. 8, pp. 231–274, 1987.

[6] Y. Brave and M. Heymann, "Control of discrete event systems modeled as hierarchical state machines", *IEEE Trans. Autom. Control*, vol. 38, pp. 1803–1819, Dec. 1993.

[7] J.S. Ostroff, *Temporal Logic for Real-Time Systems*, RSP. Research Studies Press / Wiley, 1989, Taunton, UK.

[8] M. Lawford and W.M. Wonham, "Equivalence preserving transformations for timed transition models", *in Proc. of 31st Conf. Decision and Control*, pp. 3350–3356, Tucson, AZ, USA, Dec. 1992.

[9] A. Arnold, *Finite Transition Systems*, Prentice Hall, 1994.

[10] P.C. Kanellakis and S.A. Smolka, "CCS expressions, finite state processes, and three problems of equivalence", *in Proc. of 2nd ACM Symposium on the Principles of Distributed Computing*, pp. 228–240, Montreal, Canada, Aug. 1983. ACM.

[11] R. Paige and R.E. Tarjan, "Three partition refinement algorithms", *SIAM J. of Computing*, vol. 16, pp. 973–989, Dec. 1987.

[12] B. Bolognesi and M. Caneve, "Equivalence verification: Theory, algorithms and a tool", in C. Vissers P. van Eijk and M. Diaz, editors, *The Formal Description Technique LOTOS*, pp. 303–326. North-Holland, 1989.

[13] P.J. Ramadge and W.M. Wonham, "Supervisory control of a class of discrete event processes", *SIAM J. Control Optim.*, vol. 25, pp. 206–230, Jan. 1987.