On the Computation of Observers in Discrete-Event Systems

K.C. Wong and W.M. Wonham
Systems Control Group
Dept. of Elec. and Comp. Eng.
University of Toronto
Toronto, Canada M5S 1A4
kcwong,wonham@control.utoronto.ca

Abstract — **The concept of observer was introduced in previous work by the authors on a hierarchical control theory of discrete-event systems (DES). It was shown that the observer property ensures that in a two-level hierarchy the low-level implementation of a nonblocking high-level supervisor is also nonblocking. In this paper we investigate the following problem:** *Given a causal reporter map that is not an observer, how do we design an observer by modifying the given map?* **In the case where the given causal reporter map is represented by a finite Mealy automaton, an effective computational algorithm is developed for computing an observer with the coarsest possible equivalence kernel that is finer than that of the given map. Three examples are provided for illustration.**

I. Introduction

The concept of observer was introduced in [7], where it was shown to play a useful role in the hierarchical control theory of [9] by ensuring the preservation of the nonblocking property from a high-level control synthesis to its low-level implementation. Also, it is well known that a 'natural' (zero-memory) projection of a DES might produce a more complex, even exponentially more complex, projected model. In [5], it was shown that a natural projection with the observer property always yields a projected DES that is simpler than the original DES.

In this paper we investigate the following problem: *Given a causal reporter map that is not an observer, how do we compute an observer by modifying the given map?*[1]

We outline our solution to this problem. First we note that a causal reporter map is a function from one formal language to another. It turns out that the equivalence kernel of a causal reporter map can be characterized as a 'quasi-congruence' (see Section II below) for the prefix-closure operator. An observer is a causal reporter map whose equivalence kernel is a quasi-congruence for a given postfix function. Thus, observers are characterized by the set of quasi-congruences for both the prefix and postfix functions. It turns out that these quasi-congruences form a complete upper semilattice. Thus, for a given causal map, the coarsest quasi-congruence for the prefix and postfix functions that is finer than the equivalence kernel of the given map exists. Call this quasi-congruence π^*. A natural solution to our problem would be to modify the given map in such a way that the equivalence kernel of the new function is π^*.

[1] The problem of computing an observer from a given causal reporter map has been studied in [2]. However, the procedure there differs from ours and it, unlike ours, does not generally produce an observer with the coarsest possible equivalence kernel.

The computation is carried out on a finite Mealy automaton $\mathbf{G}$ representing the given causal map. As the first step, we compute the coarsest quasi-congruence ρ on $\mathbf{G}$. Because $\mathbf{G}$ is finite, ρ can be computed in a finite number of iterations of a monotone operator. Then a modification of the map is carried out by relabeling the transitions of $\mathbf{G}$ in accordance with ρ. This process is then repeated.

As a termination criterion for the above iterative computation, a characterization of the observer property is given. A causal map is an observer if and only if the quotient automaton $\overline{\mathbf{G}}$ constructed from $\mathbf{G}$ and ρ is deterministic and contains no 'unobservable'-transitions. So, after computing ρ on $\mathbf{G}$ and constructing $\overline{\mathbf{G}}$, if $\overline{\mathbf{G}}$ is deterministic and free of unobservable transitions, then the given map is an observer and the computation terminates. If not, then relabeling is needed on $\mathbf{G}$. From the above characterization, a natural relabeling scheme would be to relabel the events so that the quotient becomes deterministic and free of unobservable transitions. After relabeling, the coarsest quasi-congruence for the new Mealy automaton can be computed, and the process repeated.

It can be shown that this iterative computation converges finitely. Furthermore, upon convergence the equivalence kernel of the new causal reporter map is π^*.

The overall complexity of this computation is $O(m^4k)$, where m is the number of states of $\mathbf{G}$ and k the number of transitions.

For illustration three examples are provided.

II. Preliminaries

In this section we recall some facts that are needed for the development of the paper.

Let Σ be a nonempty, finite alphabet of event labels and Σ^+ be the set of all finite strings of elements in Σ. Let $\Sigma^* := \Sigma^+ \cup \{\epsilon\}$, where $\epsilon (\notin \Sigma)$ denotes the empty string. A subset $L \subseteq \Sigma^*$ is called a *language* over Σ. A string $s \in L$ is a *word* of L. For $s, t \in \Sigma^*$, s is a *prefix* of t, denoted $s \leq t$, if there is a string u such that $t = su$. The closure of L is the language $\overline{L}$ consisting of all the prefixes of words in L: $\overline{L} := \{s \in \Sigma^* \mid (\exists t \in L)\, s \leq t\}$. A language L is *(prefix) closed* if $L = \overline{L}$.

Let L be a language, not necessarily closed, over Σ and let T be another event alphabet. A function $\theta : \overline{L} \longrightarrow T^*$ is a *causal reporter* map [9] if it satisfies the following properties:

$$\theta(\epsilon) = \epsilon,$$
$$\theta(s\sigma) = \begin{cases} \text{either } \theta(s) \\ \text{or } \theta(s)\tau, \text{ for some } \tau \in T \end{cases} \qquad (1)$$

for $s \in \overline{L}$ and $\sigma \in \Sigma$. In the two-level hierarchy of [9], the information channel which summarizes behaviors in the low-level process and reports relevant information to the high-level system is modeled by a causal reporter map.

We will make the following assumption. Let

$$\Sigma_l := \{\sigma \in \Sigma \,|\, (\exists s \in \Sigma^*)\, s\sigma \in \overline{L} \text{ and } \theta(s\sigma) = \theta(s)\}$$

We will assume for this paper that

$$\Sigma_l \cap T = \emptyset \tag{2}$$

The assumption is not restrictive, as Σ and T are generally disjoint. A natural projection [3] also satisfies this assumption.

It turns out that the equivalence kernel of a causal reporter map can be characterized as a 'quasi-congruence' for the prefix-closure function. Let X be a set and $\alpha : X \longrightarrow \mathcal{P}(X)$ be an arbitrary function, where $\mathcal{P}(X)$ denotes the power set of X. We denote by $\mathcal{E}(X)$ the set of all equivalence relations on X. For $\omega \in \mathcal{E}(X)$, let X/ω denote the set of equivalence classes of ω, and let $p_\omega : X \longrightarrow X/\omega$ be the canonical projection associated with ω. Define $\omega \circ \alpha \in \mathcal{E}(X)$ according to the following: for $x, x' \in X$,

$$x \equiv x' \,(mod\, \omega \circ \alpha) \iff p_\omega(\alpha(x)) = p_\omega(\alpha(x'))$$

For two equivalence relations π and ω on X, π *refines* ω, denoted $\pi \leq \omega$, if for all $x, x' \in X$, $x \equiv x' \,(mod\, \pi)$ implies $x \equiv x' \,(mod\, \omega)$. We denote the finest and coarsest possible equivalence relations on X by $\perp$ and $\top$ respectively. We say that ω is a *quasi-congruence* for (X, α) [8] if $\omega \leq \omega \circ \alpha$.[2] The equivalence relation $\perp$ is a quasi-congruence for (X, α), and the quasi-congruences for (X, α) form a complete upper semilattice under the *join* operation $\vee$ of the lattice $\mathcal{E}(X)$.

Define $pre : \Sigma^* \longrightarrow \mathcal{P}(\Sigma^*) : s \longmapsto \overline{\{s\}}$. Let $ker\, \theta$ denote the equivalence kernel of θ: for $s, s' \in \overline{L}$, $s \equiv s' \,(mod\, ker\, \theta)$ iff $\theta(s) = \theta(s')$.

Proposition II.1 ([6]). *If θ is a causal reporter map from $\overline{L}$ to T^*, then $ker\, \theta$ is a quasi-congruence for $(\overline{L}, pre)$. Conversely, for a quasi-congruence ω for $(\overline{L}, pre)$, there are an event alphabet T and a causal reporter map $\theta : \overline{L} \longrightarrow T^*$ such that $ker\, \theta = \omega$.* □

We now recall the definition of observers. For a given language H (not necessarily closed) over Σ, the *postfix of* $s \in \Sigma^*$ *in* H, denoted $pos_H(s)$, consists of all extensions of s in H:

$$pos_H(s) = \{u \in H \,|\, s \leq u\}$$

Thus, pos_H is a function from Σ^* to $\mathcal{P}(H)$.

Let $H \subseteq \overline{L}$ and θ be a causal reporter map from $\overline{L}$ to T^*. We say that θ is an H*-observer* [6] if $ker\, \theta$ is a quasi-congruence for $(\overline{L}, pos_H)$. If $H = \overline{L}$, we say that θ is an *observer.* In this case, it can be shown that our definition of an observer is logically equivalent to that in [7].

Thus, we can compute an H-observer by computing a quasi-congruence for both $(\overline{L}, pre)$ and $(\overline{L}, pos_H)$.

Let L be a language over Σ, not necessarily closed, and let $\theta : \overline{L} \longrightarrow T^*$ be a causal reporter map. If θ is not an L-observer, then our problem is to modify θ such that the resultant θ' is an L-observer. From the facts on quasi-congruences, we see that

$$\pi^* := sup\{\pi \in \mathcal{E}(\overline{L}) \,|\, \pi \leq (ker\, \theta) \wedge (\pi \circ pre) \wedge (\pi \circ pos_L)\}$$

exists, where $\wedge$ denotes the *meet* operation. We will modify θ in an iterative fashion such that the equivalence kernel of the resultant map θ' is π^*, and thus obtain a natural solution to our problem.

[2] A quasi-congruence is called a *bisimulation* in computer science. See [4].

III. Computation on finite automata

The computation is carried out on a Mealy automaton representing the reporter map. Let

$$\widehat{\mathbf{G}} = (Q, \Sigma, T \cup \{\tau_0\}, \widehat{\delta}, \phi, q_0, Q_m) \tag{3}$$

be a trim, deterministic Mealy automaton representing (L, θ), where the entries of $\widehat{\mathbf{G}}$ are respectively the state set, event alphabet, output event set, transition function, output function, initial state and the set of marker states. Here $\tau_0 \notin \Sigma \cup T$.

We introduce the following definition of relabeling. Let $\mathbf{A} = (A, \Gamma, \mu, a_0, A_m)$[3] be a nondeterministic automaton. The set of transitions of $\mathbf{A}$ is given by

$$T_\mu := \{(a, \gamma, a') \in A \times \Gamma \times A \,|\, a' \in \mu(a, \gamma)\}$$

An automaton $\mathbf{A}' = (A, \Gamma', \mu', a_0, A_m)$, possibly nondeterministic, is a *relabeling* of $\mathbf{A}$ if there is a surjective mapping $r : T_\mu \longrightarrow T_{\mu'} : (a, \gamma, a') \longmapsto (a, \gamma', a')$, where $\gamma' \in \Gamma'$. The function r is called the *relabeling function* from $\mathbf{A}$ to $\mathbf{A}'$. If $\mathbf{A}'$ is deterministic, then we say that $\mathbf{A}'$ is a *deterministic relabeling* of $\mathbf{A}$.

We incorporate the information of the output function ϕ into the transition structure by relabeling some of the events. Let

$$\mathbf{G} = (Q, \Sigma \cup T, \delta, q_0, Q_m) \tag{4}$$

be a relabeling of $(Q, \Sigma, \widehat{\delta}, q_0, Q_m)$ with the relabeling function $r_{\mathbf{G}} : T_{\widehat{\delta}} \longrightarrow T_\delta$ satisfying the following: for $(q, \sigma, q') \in T_{\widehat{\delta}}$,

$$r_{\mathbf{G}}((q, \sigma, q')) = \begin{cases} (q, \tau, q') & \text{if } \phi(q, \sigma) = \tau \in T \\ (q, \sigma, q') & \text{if } \phi(q, \sigma) = \tau_0 \end{cases} \tag{5}$$

In general, $\mathbf{G}$ is nondeterministic.

For this paper we assume that

$$\mathbf{G} \text{ is a finite transition structure.}$$

We define, for $\tau \in T$, $\Delta_\tau : Q \longrightarrow \mathcal{P}(Q)$:

$$\Delta_\tau(q) := \bigcup\{\delta(q, u\tau u') \,|\, uu' \in (\Sigma - T)^*\}$$

and $\Delta_{\epsilon,m} : Q \longrightarrow \mathcal{P}(Q_m)$:

$$\Delta_{\epsilon,m}(q) := \bigcup\{\delta(q, u) \cap Q_m \,|\, u \in (\Sigma - T)^*\}$$

Let

$$\rho_{\mathbf{G}} := sup\{\rho \in \mathcal{E}(Q) \,|\, \rho \leq (\bigwedge_{\tau \in T} (\rho \circ \Delta_\tau)) \wedge (\rho \circ \Delta_{\epsilon,m})\} \tag{6}$$

Here $\rho_{\mathbf{G}}$ is the coarsest quasi-congruence for $(Q, (\{\Delta_\tau \,|\, \tau \in T\}, \Delta_{\epsilon,m}))$ that is finer than $\top \in \mathcal{E}(Q)$.

To compute $\rho_{\mathbf{G}}$, define $\varphi : \mathcal{E}(Q) \longrightarrow \mathcal{E}(Q)$:

$$\varphi(\rho) = (\bigwedge_{\tau \in T} (\rho \circ \Delta_\tau)) \wedge (\rho \circ \Delta_{\epsilon,m})$$

It can be verified that φ is monotone and that $\rho_{\mathbf{G}}$ is the greatest fixpoint of φ.

Let

$$\rho_0 := \varphi^0(\top) = \top \text{ and } \rho_i := \varphi(\rho_{i-1})$$

[3] In this paper, the five entries in the five-tuple of an automaton are, in the order listed, the state set, event set, transition function, the initial state and the set of marker states.

for $i > 0$. Since φ is monotone and $\varphi(\top) \leq \top$, $\{\rho_i \,|\, i \geq 0\}$ is a nonincreasing sequence. Since $\mathbf{G}$ is a finite transition structure,

$$\rho_{\mathbf{G}} = inf\{\rho_i \,|\, i \geq 0\} = \rho_k$$

for some $k \geq 0$, i.e., $\rho_{\mathbf{G}}$ is attained in a finite number of steps.

This computation of $\rho_{\mathbf{G}}$ can be carried out by using the partition refinement algorithm of [1]. Assume that $m = |Q|$ and $n = |T|$. Following [1], the partition $\rho_{\mathbf{G}}$ can be computed in $O(m^3 n)$ time.

As an illustration, we consider the following.

Example III.1. Let a causal reporter map be given by the Mealy automaton in Figure 1.

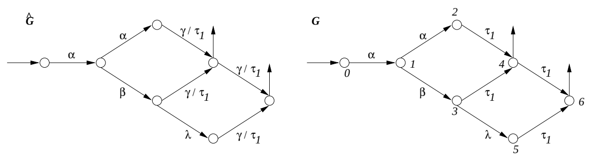

Figure 1: A Mealy automaton.

We compute $\rho_{\mathbf{G}}$. Here, $\rho_1 = \varphi(\top) = (\top \circ \Delta_{\tau_1}) \wedge (\top \circ \Delta_{\epsilon,m})$ and $\rho_2 := \varphi(\rho_1)$. They are displayed in Figure 2. In this case,

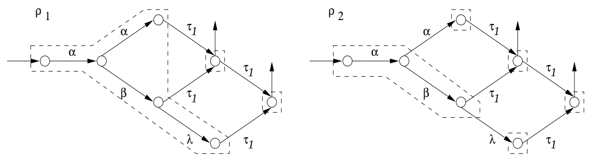

Figure 2: The partitions ρ_1 and ρ_2 on $\mathbf{G}$.

$\rho_{\mathbf{G}} = \rho_2$. $\diamond$

IV. A characterization of observers

In this section, we present a characterization of the observer property. Roughly, a causal reporter map θ is an L-observer if and only if a certain 'quotient' of a Mealy automaton for (L,θ) is deterministic and contains no 'unobservable' transition.

First we make the following definition of a quotient automaton. Let $\mathbf{A} = (A, \Gamma, \mu, a_0, A_m)$ be a nondeterministic automaton. For $\Gamma' \subseteq \Gamma$ and $A' \subseteq A$, define

$$\mu_{\Gamma'}(A') := \bigcup \{\mu(a,\gamma) \,|\, a \in A',\ \gamma \in \Gamma'\}$$

If Γ' is a singleton, i.e. $\Gamma' = \{\gamma\}$, we write μ_γ instead of $\mu_{\Gamma'}$.

For a partition ρ on A, let $B := A/\rho$ be the quotient set and $g : A \longrightarrow B$ be the corresponding canonical projection. Let $b_0 := g(a_0)$ and $B_m := g(A_m)$ be respectively the initial state and the marker states in the quotient. Let $\gamma_0 \notin \Gamma$ and $\Gamma_o \subseteq \Gamma$. Intuitively, it might be helpful to think of the labels in Γ_o as *observable* labels and those in $\Gamma - \Gamma_o$ as *unobservable* labels. Define the (induced) transition function on the quotient as follows. $\nu : B \times (\Gamma_o \cup \{\gamma_0\}) \longrightarrow \mathcal{P}(B)$

$$\nu(b,\gamma) = \begin{cases} g(\mu_\gamma(g^{-1}(b))) & \text{if } \gamma \in \Gamma_o \\ g(\mu_{(\Gamma-\Gamma_o)}(g^{-1}(b))) - \{b\} & \text{otherwise (i.e. } \gamma = \gamma_0) \end{cases}$$

Here γ_0 labels an 'unobservable' transition.

Define the *quotient* of $\mathbf{A}$ over Γ_o and ρ as

$$\mathbf{A}/(\Gamma_o,\rho) := (B, \Gamma_o \cup \{\gamma_0\}, \nu, b_0, B_m)$$

In general the quotient $\mathbf{A}/(\Gamma_o,\rho)$ might be nondeterministic.

Let

$$\mathbf{G} = (Q, \Sigma \cup T, \delta, q_0, Q_m)$$

be an automaton obtained from a relabeling of a trim Mealy automaton of (L,θ) as in (4) and (5). Let $\rho_{\mathbf{G}} \in \mathcal{E}(Q)$ be the coarsest quasi-congruence for $\mathbf{G}$ and T that is finer than $\top$ as defined in (6). Let

$$\overline{\mathbf{G}} := \mathbf{G}/(T, \rho_{\mathbf{G}}) = (Y, T \cup \{\tau_0\}, \eta, y_0, Y_m)$$

where τ_0 is an event label that is not in $\Sigma \cup T$. Let $g : Q \longrightarrow Y$ be the canonical projection associated with $\rho_{\mathbf{G}}$.

Theorem IV.1. *θ is an L-observer iff $\overline{\mathbf{G}}$ is deterministic and contains no τ_0-transition.* $\square$

The following is a direct result of the characterization in Theorem IV.1, and states that if θ is an L-observer then the minimal recognizer for the language $\theta(L)$ is always simpler than a Mealy automaton representing (L,θ).

Theorem IV.2. *(i) if θ is an L-observer, then $\overline{\mathbf{G}}$ (deterministic and free of τ_0-transitions) is a minimal-state recognizer of the language $\theta(L)$. (ii) $\overline{\mathbf{G}}$ can be computed in polynomial time.* $\square$

V. Relabeling algorithm

In this section we present a relabeling scheme. Roughly, the scheme is to relabel $\mathbf{G}$ in such a way that the quotient of the relabeled automaton over $\rho_{\mathbf{G}}$ is deterministic and contains no τ_0-transitions. We see from Theorem IV.1 that, if the quotient $\mathbf{G}$ over $\rho_{\mathbf{G}}$ is deterministic and free of τ_0-transitions, then θ is an L-observer. Thus, intuitively through relabeling we are modifying the reporter map in a direction toward an observer.

Let $\mathbf{A} = (A, \Gamma, \mu, a_0, A_m)$, possibly nondeterministic, and $\Gamma_o \subseteq \Gamma$. Let $\rho \in \mathcal{E}(A)$ and let

$$\overline{\mathbf{A}} = \mathbf{A}/(\Gamma_o,\rho) = (B, \Gamma_o \cup \{\gamma_0\}, \nu, b_0, B_m)$$

be the quotient of $\mathbf{A}$ over Γ_o and ρ.

From the definition of quotient, $B = A/\rho$. Let $g : A \longrightarrow B$ be the canonical projection. The following definitions are used in formulating the relabeling procedure. Define a map from the transitions of $\mathbf{A}$ to those of $\overline{\mathbf{A}}$: $g_t : T_\mu \longrightarrow T_\nu$

$$g_t((a,\gamma,a')) = \begin{cases} (g(a), \gamma, g(a')) & \text{if } \gamma \in \Gamma_o \\ (g(a), \gamma_0, g(a')) & \text{if } \gamma \notin \Gamma_o, \text{ and } g(a) \neq g(a') \\ \text{undefined} & \text{otherwise} \end{cases}$$

We call the function g_t the *transition mapping* from $\mathbf{A}$ to $\mathbf{A}/(\Gamma_o,\pi)$.

Let $D_{g_t}(T_\mu)$ denote the set of transitions on which g_t is defined, namely

$$D_{g_t}(T_\mu) := \{\tau \in T_\mu \,|\, g_t(\tau) \text{ is defined }\}$$

As discussed above, the goal of this procedure is to relabel $\mathbf{A}$ to make the quotient $\overline{\mathbf{A}}$ deterministic and free of γ_0-transitions. So, to achieve this we first form the quotient $\overline{\mathbf{A}}$, then relabel $\overline{\mathbf{A}}$ so that it is deterministic and γ_0-transition free, and finally relabel $\mathbf{A}$ in accordance with the relabeling of the quotient. These are the three basic steps in the relabeling procedure.

We now present the procedure.

Algorithm V.1 (Relabeling).
Input: $\mathbf{A} = (A, \Gamma, \mu, a_0, A_m)$ *is a (possibly nondeterministic) automaton,* Γ_o *is a subset of* Γ*, and* ρ *is a partition on* A.

1. *Compute* $\overline{\mathbf{A}} := \mathbf{A}/(\Gamma_o, \rho) = (B, \Gamma_o \cup \{\gamma_0\}, \nu, b_0, B_m)$.

2. *Let* $\overline{\mathbf{A}}' = (B, \Gamma'_o, \nu', b_0, B_m)$ *be a deterministic relabeling of* $\overline{\mathbf{A}}$ *such that* $\gamma_0 \notin \Gamma'_o$ *and the relabeling function* $\overline{r} : T_\nu \longrightarrow T_{\nu'}$ *satisfies the following.*

   $(i)(\forall \tau = (b, \gamma, b') \in T_\nu)\, \overline{r}(\tau) \neq \tau \Longrightarrow (\exists \gamma' \in \Gamma'_o - \Gamma)\, \overline{r}(\tau) = (b, \gamma', b')$

   (All relabelings are done with new labels.)

   $(ii)(\forall (b_1, \gamma, b_2), (b'_1, \gamma', b'_2) \in T_\nu)(\forall \beta \in \Gamma'_o)$ $\overline{r}((b_1, \gamma, b_2)) = (b_1, \beta, b_2),\ \overline{r}((b'_1, \gamma', b'_2)) = (b'_1, \beta, b'_2) \Longrightarrow \gamma = \gamma'$

   (If two transitions are relabeled with same label then the two original labels are the same.)

   (It can be checked that such a relabeling is well defined.)

3. *Let* $\mathbf{A}' = (A, \Gamma', \mu', a_0, A_m)$ *be a relabeling of* $\mathbf{A}$ *such that* $\Gamma'_o \subseteq \Gamma'$ *and the relabeling function* $r : T_\mu \longrightarrow T_{\mu'}$ *satisfies the following.*

   $(i)(\forall \tau \in T_\mu - D_{g_t}(T_\mu))\, r(\tau) = \tau,$ *and*

   (The transitions with unobservable labels and with both states in the same coset will not be relabeled.)

   $(ii)(\forall \tau = (a, \gamma, a') \in D_{g_t}(T_\mu))(\exists \gamma' \in \Gamma'_o)$ $r((a, \gamma, a')) = (a, \gamma', a')$ *and* $(\overline{r} \circ g_t)(\tau) = (g'_t \circ r)(\tau)$

   (Other transitions are relabeled in accordance with $\overline{r}$*.)*

   where g'_t *is the transition mapping from* $\mathbf{A}'$ *to* $\mathbf{A}'/(\Gamma'_o, \rho)$.

Output: $\mathbf{A}' = (A, \Gamma', \mu', a_0, A_m)$*, possibly nondeterministic, and* $\Gamma'_o \subseteq \Gamma'$. ◇

If we apply Algorithm V.1 to a finite $\mathbf{A}$, then it is clear that the algorithm terminates.

As a convenient notation, we write

$$(\mathbf{A}', \Gamma'_o) = relabel_\rho(\mathbf{A}, \Gamma_o)$$

to indicate an application of the relabeling algorithm.

In Step 2 of the relabeling algorithm, the set of event labels in $\overline{\mathbf{A}}'$ is Γ'_o which does not contain γ_0. We interpret this set Γ'_o as the set of observable event labels in the relabeled system. Thus the relabeling operation which produces $\overline{\mathbf{A}}'$ from $\overline{\mathbf{A}}$ eliminates γ_0-transitions. Since $\overline{\mathbf{A}}'$ is a deterministic relabeling of $\overline{\mathbf{A}}$, the relabeling operation also eliminates nondeterminism.

The property in $2(i)$ requires that any relabeling be carried out with new event labels (i.e. labels that are not in Γ). With this the assumption (2) remains satisfied in the relabeled system. Note that the observable events that have not been relabeled remain observable in the relabeled system.

Property $2(ii)$ states that if, after relabeling, two transitions have the same event label, then the original event labels (i.e. those before relabeling) are also the same. If $(b_1, \gamma, b_2), (b'_1, \gamma', b'_2) \in T_\nu$ and $\gamma \neq \gamma'$, then after relabeling these transitions will again have different labels. Thus the relabeling scheme preserves the initial partition of events.

In Step 3, $\mathbf{A}$ is relabeled to produce an $\mathbf{A}'$ with the property that the quotient of $\mathbf{A}'$ over Γ'_o and ρ is $\overline{\mathbf{A}}'$. Property $3(i)$ states that any transition with an unobservable event label and with its exit and entrance states in the same equivalence class of ρ will not be relabeled. For other transitions, according to $3(ii)$ they are relabeled in accordance with the relabeling of their g_t-images. Let $\tau = (a, \gamma, a') \in D_{g_t}(T_\mu)$. Then by $3(ii)$ $r((a, \gamma, a')) = (a, \gamma', a')$, for some $\gamma' \in \Gamma'_o$. Suppose that $\overline{r}$ relabels $g_t(\tau) = (g(a), \gamma, g(a'))$ to $(g(a), \gamma'', g(a'))$, for some $\gamma'' \in \Gamma'_o$. That is, $(\overline{r} \circ g_t)(\tau) = (g(a), \gamma'', g(a'))$. Thus,

$$\begin{aligned}
(g'_t \circ r)(\tau) &= (g'_t \circ r)((a, \gamma, a')) \\
&= g'_t((a, \gamma', a')) \qquad r \text{ relabels } \gamma \text{ with } \gamma' \\
&= (g(a), \gamma', g(a')) \\
&= (\overline{r} \circ g_t)(\tau) \\
&= (g(a), \gamma'', g(a'))
\end{aligned}$$

Thus $\gamma' = \gamma''$, i.e. τ is relabeled in accordance with the relabeling in $\overline{\mathbf{A}}$.

VI. Observer computation algorithm

In this section, we present the algorithm for computing an observer with the coarsest equivalence kernel that is finer than that of a given map.

Algorithm VI.1 (Observer computation).

Input: $\mathbf{G} = (Q, \Sigma \cup T, \delta, q_0, Q_m)$ *is an automaton obtained from a Mealy automaton of* (L, θ) *by relabeling as in (4) and (5).*

1. *Compute* ρ*, the coarsest quasi-congruence for* $(\mathbf{G}, T)$.
2. *Compute* $\overline{\mathbf{G}} = \mathbf{G}/(T, \rho)$.
3. *If* $\overline{\mathbf{G}}$ *is deterministic and contains no* τ_0*-transition then*
   $\mathbf{G}' = \mathbf{G}$
   $T' = T$
   Stop

   else
   $(\mathbf{G}'', T'') = relabel_\rho(\mathbf{G}, T)$
   $\mathbf{G} = \mathbf{G}''$
   $T = T''$
   Return to Step 1

   end if.

Output: $\mathbf{G}' = (Q, \Sigma \cup T', \delta', q_0, Q_m)$*, possibly nondeterministic.* ◇

Let $\mathbf{G}_0 = \mathbf{G}$ and $T_0 = T$. For $i \geq 1$, let $\rho_{i,Q}$ be the coarsest quasi-congruence for $(\mathbf{G}_{i-1}, T_{i-1})$, and let

$$(\mathbf{G}_i, T_i) = relabel_{\rho_{i,Q}}(\mathbf{G}_{i-1}, T_{i-1})$$

where $\mathbf{G}_i = (Q, \Sigma \cup T_i, \delta_i, q_0, Q_m)$.

For the i'th reporter map, where $i \geq 0$, let

$$\widehat{\mathbf{G}}_i = (Q, \Sigma, T_i \cup \{\tau_0\}, \widehat{\delta}, \phi_i, q_0, Q_m)$$

where $\widehat{\delta}$ is the same transition function as in (3), and $\phi_i : Q \times \Sigma \longrightarrow T_i \cup \{\tau_0\}$ is defined by the following.

$$\phi_i(q, \sigma) = \begin{cases} \tau & \text{if } (r^i_{\mathbf{G}} \circ \cdots \circ r^1_{\mathbf{G}} \circ r_{\mathbf{G}})(q, \sigma, \widehat{\delta}(q, \sigma)) \\ & \qquad = (q, \tau, \widehat{\delta}(q, \sigma)) \\ \tau_0 & \text{if } (r^i_{\mathbf{G}} \circ \cdots \circ r^1_{\mathbf{G}} \circ r_{\mathbf{G}})(q, \sigma, \widehat{\delta}(q, \sigma)) \\ & \qquad = (q, \sigma, \widehat{\delta}(q, \sigma)) \end{cases}$$

where $r_{\mathbf{G}}$ is given in (5) and $r^i_{\mathbf{G}}$ is the relabeling function from $(\mathbf{G}_{i-1}, T_{i-1}, \rho_{i,Q})$ to $(\mathbf{G}_i, T_i)$.

For $i \geq 0$, let $\theta_i : \overline{L} \longrightarrow T_i^*$ be the causal reporter map represented by the Mealy automaton $\widehat{\mathbf{G}}_i$.

Theorem VI.1. *Under the foregoing assumptions, Algorithm VI.1 terminates in at most $|Q|$ number of steps. If the algorithm stops after n iterations, where $n \geq 1$, then $ker\, \theta_{n-1} = \pi^*$.* □

Let $m = |Q|$ be the number of states. Recall that $\widehat{\delta}$ is the deterministic transition function of the Mealy automaton $\widehat{\mathbf{G}}$. The set of transitions of $\widehat{\mathbf{G}}$ is given by $T_{\widehat{\delta}} = \{(q, \sigma, q') \,|\, \delta(q, \sigma) = q'\}$. Let $k = |T_{\widehat{\delta}}|$. It is clear that $|T_i| \leq k$. Thus $\rho_{i,Q}$ can be computed in $O(m^3 k)$ time. Since the algorithm converges in at most m steps, the overall complexity is $O(m^4 k)$.

VII. Examples

In this section, we illustrate the algorithms with three examples.

Example VII.1. Let $\mathbf{G}$ be the automaton obtained from the Mealy automaton in Figure 1 in Section III. Let $\mathbf{G}_0 = \mathbf{G}$ and $T_0 = T$. In the first iteration of Algorithm VI.1, the coarsest quasi-congruence $\rho_{1,Q}$ is computed for $(\mathbf{G}_0, T_0)$ in Step 1. This has already been done in Example III.1; $\rho_{1,Q}$ equals $\rho_{\mathbf{G}} (= \rho_2)$ in Figure 2. In Step 2 of Algorithm VI.1, $\overline{\mathbf{G}}_0 := \mathbf{G}_0/(T_0, \rho_{1,Q})$ is computed and is displayed in Figure 3. In Step 3, since $\overline{\mathbf{G}}_0$ contains τ_0-transitions, the relabeling algorithm (Algorithm V.1) is applied to $(\mathbf{G}_0, T_0, \rho_{1,Q})$ to obtain $\mathbf{G}_1$ and T_1. Here $T_1 = \{\tau_1, \tau_2, \tau_3\}$. The automaton $\mathbf{G}_1$ is also displayed in Figure 3. Note that the quotient $\mathbf{G}_1/(T_1, \rho_{1,Q})$ is the same as $\overline{\mathbf{G}}_0$

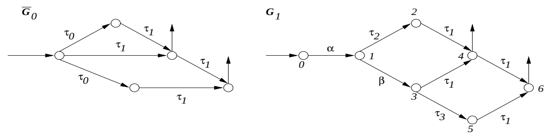

Figure 3: $\overline{\mathbf{G}}_0$ and $\mathbf{G}_1$.

except that the τ_0-transitions have been relabeled with τ_2 and τ_3.

For the second iteration of the algorithm, the coarsest quasi-congruence $\rho_{2,Q}$ for $(\mathbf{G}_1, T_1)$ is computed and is displayed in Figure 4. It can be verified that $\overline{\mathbf{G}}_1 (:= \mathbf{G}_1/(T_1, \rho_{2,Q}))$ contains a τ_0-transition. So the relabeling algorithm is used to relabel $\mathbf{G}_1$ to produce $(\mathbf{G}_2, T_2)$. Here $T_2 = \{\tau_1, \tau_2, \tau_3, \tau_4\}$. The automaton $\mathbf{G}_2$ is displayed in Figure 4.

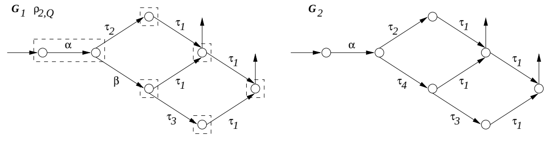

Figure 4: $(\mathbf{G}_1, \rho_{2,Q})$ and $\mathbf{G}_2$.

For the third iteration of the algorithm, we have that $\overline{\mathbf{G}}_2 := \mathbf{G}_2/(T_2, \rho_{3,Q})$ is deterministic and contains no τ_0-transition, where $\rho_{3,Q}$ is the coarsest quasi-congruence for $(\mathbf{G}_2, T_2)$. With this the computation terminates. The output of the algorithm is $\mathbf{G}_2$.

The Mealy automaton for the new causal map θ', and the automaton recognizing $\theta'(L_m)$ where L_m is the marked behavior of $\widehat{\mathbf{G}}$ in Figure 1, are displayed in Figure 5. One can verify that the equivalence kernel of the new causal map is indeed π^* ◇

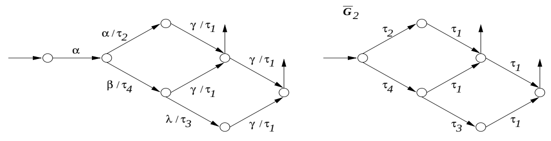

Figure 5: The Mealy automaton representing θ' and the automaton for $\theta'(L_m)$.

Example VII.2. Figure 6 displays the automaton obtained from the Moore automaton in Figure 5.6.5 in [8]. To avoid clutter, the low-level event labels are not displayed as they do not enter into the computation. Let $\mathbf{G}$ be the automaton in

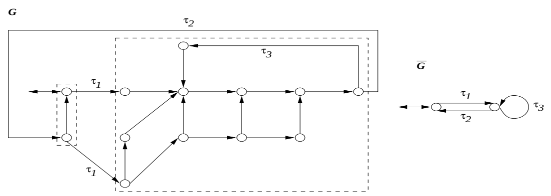

Figure 6: The results of computation.

Figure 6. Here $T = \{\tau_1, \tau_2, \tau_3\}$.

The coarsest quasi-congruence $\rho_{1,Q}$ for $(\mathbf{G}, T)$ is shown on $\mathbf{G}$ in Figure 6. The quotient automaton $\overline{\mathbf{G}} := \mathbf{G}/(T, \rho_{1,Q})$ is computed and is also displayed in Figure 6. Since $\overline{\mathbf{G}}$ is deterministic and contains no τ_0-transition, the computation terminates and the given causal map is an L_m-observer, where L_m is the marked behavior of the original Moore automaton. According to Theorem IV.2, $\overline{\mathbf{G}}$ recognizes the language $\theta(L_m)$.

The same can be applied to the Moore automaton in Figure 5.6.7 in [8] and it turns out that the given causal map is also an L_m-observer. The results of the computation are displayed in Figure 7. The automaton $\overline{\mathbf{G}}$ recognizes the language $\theta(L_m)$. ◇

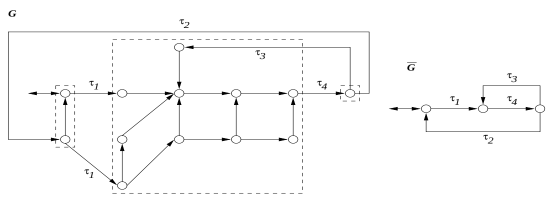

Figure 7: The results of computation for the second Moore automaton.

Example VII.3. The Moore automaton in Figure 8 is borrowed from [2]. To avoid clutter, the low-level event labels are

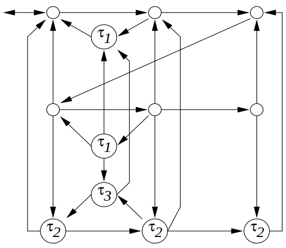

Figure 8: A Moore automaton.

again not displayed.

We relabel the Moore automaton in Figure 8 as in Section III. The result is displayed in Figure 9. Let $\mathbf{G}_0$ and T_0 be respectively the automaton in Figure 9 and the high-level event set. In the first iteration of the observer computation

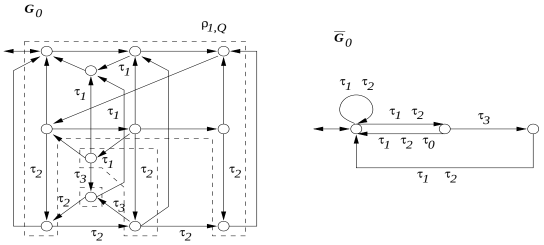

Figure 9: The automaton $\mathbf{G}_0$, the partition $\rho_{1,Q}$ and the quotient $\overline{\mathbf{G}}_0$.

algorithm, the coarsest quasi-congruence $\rho_{1,Q}$ is computed and is shown in Figure 9. The quotient $\overline{\mathbf{G}}_0 := \mathbf{G}_0/(T,\rho_{1,Q})$ is also displayed in Figure 9. To eliminate nondeterminism and τ_0-transition, $\mathbf{G}_0$ is relabeled to produce $\mathbf{G}_1$ which is shown in Figure 10. Here the high-level event set $T_1 =$

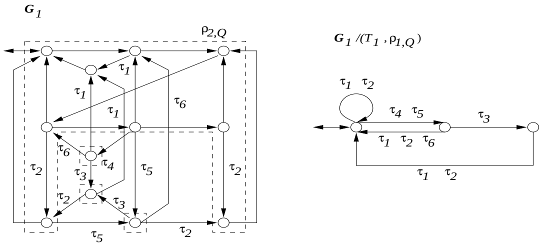

Figure 10: The automata $\mathbf{G}_1$ and $\mathbf{G}_1/(T_1,\rho_{1,Q})$.

$\{\tau_1,\tau_2,\tau_3,\tau_4,\tau_5,\tau_6\}$. The quotient $\mathbf{G}_1/(T_1,\rho_{1,Q})$ is also displayed in Figure 10, which shows that the relabeling indeed removes nondeterminism and the τ_0-transition.

In the second iteration, the coarsest quasi-congruence $\rho_{2,Q}$ is computed and is displayed in Figure 10. The quotient $\overline{\mathbf{G}}_1 := \mathbf{G}_1/(T_1,\rho_{2,Q})$ is presented in Figure 11. It is clear that it is deterministic and contains no τ_0-transitions. Thus the computation terminates and the result is $\mathbf{G}_1$ in Figure 10. From $\mathbf{G}_1$, we obtain the Moore automaton given in Figure 12.

According to our theory, this new causal map represented by the Moore automaton in Figure 12 is an L_m-observer,

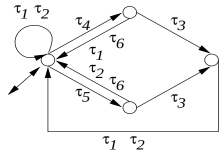

Figure 11: The automaton $\overline{\mathbf{G}}_1 = \mathbf{G}_1/(T_1,\rho_{2,Q})$.

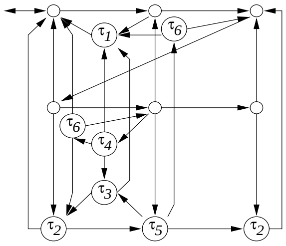

Figure 12: The Moore automaton for the new causal map.

where L_m is the marked behavior of $\mathbf{G}$. According to Theorem IV.2, $\overline{\mathbf{G}}_1$ recognizes the language $\theta_1(L_m)$. $\diamond$

VIII. Conclusions

In the case where a given causal reporter map is represented by a finite Mealy automaton, an effective computational algorithm has been developed, for computing an observer with the coarsest equivalence kernel that is finer than that of a given reporter map. The issues of efficient implementation of the algorithm, as well as the problem of minimal extension of the alphabet in the relabeling process, remain to be investigated.

References

[1] J.-C. Fernandez, "An implementation of an efficient algorithm for bisimulation equivalence," *Science of Computer Programming*, vol. 13, pp. 219-236, 1990.

[2] Y. Guan, *Implementation of Hierarchical Observer Theory*, M.A.Sc. thesis, Department of Electrical and Computer Engineering, University of Toronto, 1997.

[3] F. Lin, W. M. Wonham, "Decentralized supervisory control of discrete-event systems," *Information Sciences*, vol. 44, pp. 199-224, 1988.

[4] R. Milner, *Communication and Concurrency*, Prentice Hall, 1989.

[5] K. C. Wong, "On the complexity of projections of discrete-event systems," *IEE Workshop on Discrete Event Systems*, pp. 201-206, August 1998.

[6] K. C. Wong, J. G. Thistle, R. P. Malhamé, H.-H. Hoang, "Supervisory Control of Distributed Systems: Conflict Resolution," *Journal of Discrete Event Dynamic Systems: Theory and Applications.* To appear.

[7] K. C. Wong, W. M. Wonham, "Hierarchical Control of Discrete-Event Systems," *Discrete Event Dynamic Systems: Theory and Applications*, vol. 6, no. 3, pp. 241-273, 1996.

[8] W. M. Wonham, *Notes on Control of Discrete-Event Systems*, ECE 1636F/1637S, Department of Electrical and Computer Engineering, University of Toronto, 1999.

[9] H. Zhong, W. M. Wonham, "On the consistency of hierarchical supervision in discrete-event systems," *IEEE Transactions on Automatic Control*, vol. 35, no. 10, pp. 1125-1134, 1990.