Why Event Observation: Observability Revisited

CHANGWEI CAO
Delco Electronics Corporation, Kokomo, IN 46904

FENG LIN
Department of Electrical and Computer Engineering, Wayne State University, Detroit, MI 48202

ZHENG-HUI LIN
Institute of VLSI, Shanghai Jiao Tong University, Shanghai 200030, China

Received August 4, 1994; Revised April 18, 1996; Accepted July 22, 1996

Abstract. In this paper, we revisit observability defined in Lin and Wonham (1988a). We show that although the original definition is intended for event observation with projection, its significance goes beyond its original intent. Specifically, we show that (1) supervisory control with state observation can be studied in terms of event observation; (2) supervisory control problems with event mask can be converted to problems with event projection; and (3) nondeterminism is closely related to unobservability.

Keywords: Observability, state observation, event mask, nondeterminism

1. Introduction

Since it was introduced and studied by Lin and Wonham about ten years ago (Lin and Wonham 1988a, 1988b, 1990), observability has played an important role in supervisory control of discrete event systems. Observability was initially defined for event observation: We distinguish, based on practical constraints, a subset of observable events from all events. Two strings look the same if they have the same sequence of observable events. Observability requires that if two strings look the same, then they must be consistent in the sense that no conflict of one event continues after one string but not in the other should occur. This definition captures precisely the existence condition for a supervisor controlling a discrete event system with partial (event) observation (Lin and Wonham 1988a). Because of its importance, observability has been studied extensively in literature. For example, it was shown that observability is preserved under the union of languages but is not preserved under the intersection (Lin and Wonham 1988a). Algorithms for calculating the infimal observable superlanguage of a given language and maximal observable sublanguages of a given language were also developed (Ben Hadj-Alouane, Lafortune and Lin 1993, Cho and Marcus 1989, Rudie and Wonham 1990, Kumar 1993). These studies have led to solutions of several important problems in discrete event systems such as supervisory control and observation problem (SCOP) (Lin and Wonham 1988a).

As indicated, the definition of observability was initially intended for event observation only. No state observation or other observation was explicitly considered. At the first glance, this may be viewed as a drawback of the definition. However, as we will show, this
is not the case. In fact, we will demonstrate that the problem with state observation can be reformulated and solved in the context of event observation.

A similar supervisory control problem was studied in Cieslak et al. (1988), where the event observation is not taken as event projection as in Lin and Wonham (1988a), but rather as event mask. In this paper, we will formally establish the equivalence of these two approaches in terms of their expressive power.

Recently, there has been a renewed interest in the control of nondeterministic discrete event systems (Heymann 1990, Heymann and Meyer 1991, Kumar and Shayman 1993, Overkamp 1994, Shayman and Kumar 1995). We demonstrate the relationship between nondeterminism and observability and show how observability can be used in studying nondeterminism.

The main contributions of this paper are summarized as follows:

- We reformulate the state observation problem and show that it can be solved in the context of event observation (Section 2).
- We reformulate the event mask problem and show that it can be made equivalent to the event projection problem (Section 3).
- We investigate the relationship between observability and nondeterminism. We show that the system obtained by internalizing the unobservable events is (weakly) nondeterministic if and only if the original system is unobservable.

We therefore conclude that the significance of the original definition of observability goes beyond its original intent. Several problems currently under investigation can be converted into problems of observability.

2. State Observation

Let us start with the supervisory control problem with state (and event) observation: Given a discrete event system modeled by an automaton

\[ G = (\Sigma, Q, \delta, q_0, Q_m) \]

where, as usual, \( \Sigma \) is the set of events; \( Q \) the set of states; \( \delta : \Sigma \times Q \to Q \) the transition function; \( q_0 \) the initial state; and \( Q_m \) the set of marker states. In general, \( \delta \) is only a partial function. We use \( \delta(\sigma, q) \) to denote that \( \delta(\sigma, q) \) is defined. We also denote the languages generated and marked by \( G \) as \( L(G) \) and \( L_m(G) \) respectively. Both are subsets of \( \Sigma^* \), the set of all strings over \( \Sigma \), including the empty string \( \epsilon \).

Our goal is to design a supervisor so that the supervised discrete event systems achieve a control objective described by a language \( K \).

As usual, a supervisor can control (disable) events in a controllable event set \( \Sigma_c \) and can observe events in an observable event set \( \Sigma_o \). We denote \( \Sigma_{uc} = \Sigma - \Sigma_c \) and \( \Sigma_{uo} = \Sigma - \Sigma_o \). In addition to event observation, we assume that a supervisor also knows some information on the states of \( G \): It observes \( y = h(q) \), where \( h : Q \to Y \) is an output mapping from the state set \( Q \) to the output set \( Y \).
To formally define a supervisor with event and state observation, we extend the observation
(projection) \( P : \Sigma^* \rightarrow \Sigma^*_o \) to include the state observation as follows: For a given string
generated by \( G \), there is a unique sequence of states visited by the string. The supervisor
observes the corresponding state output as well as the observable events. Such observations
are characterized by the extended projection \( P_e : L(G) \rightarrow 2^{(\Sigma \cup \{e\} \times Y)}^* \) defined as

\[
P_e(\epsilon) = (\epsilon, h(q_0)) (\epsilon, h(q_0))^*
\]

\[
P_e(s\sigma) = P_e(s) (P(s), h(\delta(s\sigma, q_0))) (\epsilon, h(\delta(s\sigma, q_0)))^*.
\]

The intuition behind the definition of \( P_e \) is as follows: We imagine that the output is
constantly “flashed” to the supervisor. With each flash, a pair \((\alpha, y) \in (\Sigma \cup \{\epsilon\} \times Y)\) is
added to the observation sequence. If nothing changes between two flashes, the pair \((\epsilon, y)\)
will be added, which is described by the \(^*\) in the definition.

Since \( P_e(s) \) is the observation by a supervisor when \( s \) occurs in \( G \), a supervisor is character-
ized by a mapping (the subscript \( s \) for state observation)

\[
\gamma_s : P_e(L(G)) \rightarrow 2^\Sigma.
\]

Here \( \gamma_s(P_e(s)) \) is interpreted as the set of events enabled by \( \gamma_s \) after observing \( P_e(s) \).
Because uncontrollable elements cannot be disabled, we require \( \Sigma_{uc} \subseteq \gamma_s(P_e(s)) \). Because
of the \(^*\) in the definition, the observation sequence \( P_e(s) \) is no longer unique. But the control
must be unique. Therefore, we require

\[
(\forall t, t' \in P_e(s)) \gamma_s(t) = \gamma_s(t').
\]

We also use \( \gamma_s \) to denote the supervisor characterized by \( \gamma_s \). The language generated by
\( \gamma_s \), \( L(G, \gamma_s) \), is defined in the usual way:

(i) \( \epsilon \in L(G, \gamma_s) \)

(ii) \((\forall s \in L(G, \gamma_s)) s\sigma \in L(G, \gamma_s) \Leftrightarrow s\sigma \in L(G) \land \sigma \in \gamma_s(P_e(s)) \).

The language marked by \( \gamma_s \) is defined as

\[
L_m(G, \gamma_s) = L(G, \gamma_s) \cap L_m(G).
\]

\( \gamma_s \) is said to be nonblocking if \( L_m(G, \gamma_s) = L(G, \gamma_s) \).

Our goal is to synthesize a nonblocking supervisor \( \gamma_s \) such that \( L_m(G, \gamma_s) = K \).

To find an existence condition for such a supervisor \( \gamma_s \), we modify \( G \) by extending its
states and events as follows:

\[
G_e = (\Sigma \cup Y, Q \cup Q', \delta_e, q_0, Q_m),
\]

where \( Q' \) is the “double” of \( Q \), that is, for each \( q \in Q \), there is a double \( q' \) corresponding
to \( q \) and \( Q' \) is the set of all doubles; \( \delta_e : (\Sigma \cup Y) \times (Q \cup Q') \rightarrow (Q \cup Q') \) is the extended
transition function defined as follows:

\[
\delta_e(\sigma, q) = \delta(\sigma, q) \quad \text{if } \sigma \in \Sigma, q \in Q
\]

\[
\delta_e(y, q) = q \quad \text{if } y = h(q), q \in Q
\]

\[
\delta_e(y, q') = q \quad \text{if } y = h(q), q' \in Q',
\]
and everything else is undefined. This modification is illustrated in Figure 1. The intuition behind the construction of $G_e$ is as follows: Since the output $y = h(q)$ is now available, the supervisor is required to take this information into consideration whenever necessary. To make sure this is the case, we insert at least one output $y$ before each occurrence of event (more than one $y$ is allowed but it will make no difference since it is redundant). This will be clear after we examine the language generated by $G_e$. We also note that a different construction to translate state observation to event observation for diagnosis was proposed in Sampath et al. (1994).

To find the language generated and marked by $G_e$, let us define $e: \Sigma^* \rightarrow 2^{(\Sigma \cup Y)^*}$ as follows:

$$e(\epsilon) = h(q_0)h(q_0)^*$$

$$e(s\sigma) = e(s)\sigma h(\delta(s, q_0))h(\delta(s, q_0))^*.$$  

We can now prove the following lemma.

**Lemma 1** If $L(G)$ is not empty, then

(i) $L(G_e) = \overline{e(L(G))}$

(ii) $T_e L(G_e) = L(G)$

(iii) $L_m(G_e) = \overline{e(L_m(G))}.$

where $T_e: (\Sigma \cup Y)^* \rightarrow \Sigma^*$ is the projection.

**Proof:** (i) We prove $L(G_e) = \overline{e(L(G))}$ by an induction on the length of strings.

**Base:** By the definitions, $\epsilon \in L(G_e)$ and $\epsilon \in \overline{e(\epsilon)} \subseteq \overline{e(L(G))}$. Therefore,

$$\epsilon \in L(G_e) \iff \epsilon \in \overline{e(L(G))}.$$  

**Induction Step:** Assume that for all $w \in (\Sigma \cup Y)^*$ with $|w| \leq n$,

$$w \in L(G_e) \iff w \in \overline{e(L(G))}.$$
Then for \( y \in Y \),

\[
\begin{align*}
wy &\in L(G_e) \\
\Leftrightarrow w &\in L(G_e) \land wy \in L(G_e) \\
&\Leftrightarrow w \in e(L(G)) \land wy \in L(G_e) \\
&\Leftrightarrow w \in e(L(G)) \land \delta_e(y, \delta_e(w, q'_o))! \\
&\Leftrightarrow (\exists s \in L(G)) w \in e(s) \land \delta_e(w, q'_o) = \delta(s, q_o) \land y = h(\delta(s, q_o)) \\
&\Leftrightarrow (\exists s \in L(G)) wy \in e(s) \\
&\Rightarrow wy \in e(L(G)).
\end{align*}
\]

For \( \sigma \in \Sigma \),

\[
\begin{align*}
w\sigma &\in L(G_e) \\
\Leftrightarrow w &\in L(G_e) \land w\sigma \in L(G_e) \\
&\Leftrightarrow w \in L(G) \land w\sigma \in L(G_e) \\
&\Leftrightarrow w \in e(L(G)) \land \delta_e(\sigma, \delta_e(w, q'_o))! \\
&\Leftrightarrow (\exists s \in L(G)) w \in e(s) \land \delta_e(w, q'_o) = \delta(s, q_o) \land \delta(w\sigma, q_o)! \\
&\Leftrightarrow (\exists s\sigma \in L(G)) w\sigma \in e(s\sigma) \\
&\Rightarrow w\sigma \in e(L(G)).
\end{align*}
\]

(ii) Clearly \( T_e(e(s)) = [s] \) for all \( s \in L(G) \). Therefore,

\[
T_e(L(G_e)) = T_e(e(L(G))) = L(G).
\]

(iii) Because \( L(G_e) = e(L(G)) \) and the marker states of \( G_e \) are defined to be \( Q_m, L_m(G_e) = e(L_m(G)) \). □

Since our goal is to find an existence condition for \( \gamma_e \) based on \( G_e \), let us extend the desired language \( K \) to \( K_e \) as

\[
K_e = T_e^{-1}K \cap L_m(G_e).
\]

**LEMMA 2** Assume that \( K = K \cap L_m(G) \). Then

(i) \( \overline{K_e} = T_e^{-1}K \cap L(G_e) \)

(ii) \( K_e = \overline{K_e} \cap L_m(G_e) \)

(iii) \( T_eK_e = \overline{K} \).

**Proof:** (i) (\( \subseteq \)):

\[
K_e = T_e^{-1}K \cap L_m(G_e) \\
\subseteq T_e^{-1}K \cap L_m(G_e) \\
= T_e^{-1}K \cap L(G_e).
\]
(2):

\[ w \in T_e^{-1} \overline{K} \cap L(G_e) \]
\[ \Rightarrow w \in L(G_e) \land T_e w \in \overline{K} \]
\[ \Rightarrow w \in e(L(G)) \land T_e w \in \overline{K} \]
\[ \Rightarrow (\exists s \in L(G)) w \in e(s) \land T_e w = s \in \overline{K} \]
\[ \Rightarrow (\exists s \in L(G)) w \in e(s) \land (\exists t \in \Sigma^*) s t \in \overline{K} \]
\[ \Rightarrow (\exists s \in L(G)) w \in e(s) \land (\exists t \in \Sigma^*) s t \in \overline{K} \cap L_m(G) \]
\[ \Rightarrow (\exists s \in L(G)) (\exists t \in \Sigma^*) (\exists v \in (\Sigma \cup Y)*) w v \in e(s t) \land s t \in L_m(G) \land s t \in \overline{K} \]
\[ \Rightarrow (\exists s \in L(G)) (\exists t \in \Sigma^*) (\exists v \in (\Sigma \cup Y)*) w v \in e(s t) \land w v \in e(L_m(G)) \land s t \in \overline{K} \]
\[ \Rightarrow (\exists s \in L(G)) (\exists t \in \Sigma^*) (\exists v \in (\Sigma \cup Y)*) w v \in e(s t) \land w v \in L_m(G_e) \]
\[ \land T_e (w v) \in \overline{K} \]
\[ \Rightarrow (\exists v \in (\Sigma \cup Y)*) w v \in T_e^{-1} \overline{K} \cap L_m(G_e) \]
\[ \Rightarrow w \in \overline{K_e}. \]

(ii)

\[ \overline{K_e} \cap L_m(G_e) = T_e^{-1} \overline{K} \cap L(G_e) \cap L_m(G_e) \]
\[ = T_e^{-1} \overline{K} \cap L_m(G_e) \]
\[ = \overline{K_e}. \]

(iii) (\subseteq):

\[ T_e \overline{K_e} = T_e (T_e^{-1} \overline{K} \cap L(G_e)) \]
\[ \subseteq T_e (T_e^{-1} \overline{K}) \cap T_e L(G_e) \]
\[ = \overline{K} \cap L(G) \]
\[ = \overline{K}. \]

(2):

\[ s \in \overline{K} \]
\[ \Rightarrow s \in \overline{K} \land s \in L(G) \]
\[ \Rightarrow s \in \overline{K} \land e(s) \subseteq e(L(G)) \]
\[ \Rightarrow s \in \overline{K} \land e(s) \subseteq L(G_e) \]
\[ \Rightarrow (\exists w \in e(s)) T_e w = s \in \overline{K} \land w \in L(G_e) \]
\[ \Rightarrow (\exists w \in e(s)) w \in T_e^{-1} \overline{K} \cap L(G_e) \]
\[ \Rightarrow (\exists w \in e(s)) w \in \overline{K_e} \]
\[ \Rightarrow s \in T_e \overline{K_e}. \]

Now an existence condition for \( y_e \) can be expressed in terms of controllability and observability of \( K_e \), which are defined in the usual way as follows. (Naturally, we assume that the artificial events in \( Y \) are uncontrollable).

**Definition 1**  \( K_e \) is controllable (with respect to \( L(G_e) \) and \( \Sigma_u \)) if

\[ (\forall w \in \overline{K_e}) (\forall \sigma \in \Sigma_u)(w \sigma \in L(G_e) \Rightarrow w \sigma \in \overline{K_e}). \]
To simplify the notation we also use $P$ to denote the projection from $(\Sigma \cup Y)^*$ to $(\Sigma_a \cup Y)^*$, which is consistent with the original definition.

**Definition 2** $K_e$ is observable (with respect to $L(G_e)$ and $P$) if

$$\forall w, w' \in (\Sigma \cup Y)^* (Pw = Pw' \implies \text{consis}(w, w')),$$

where $\text{consis}(w, w')$ is true if and only if

$$\forall \sigma \in (\Sigma \cup Y) (w \sigma \in \overline{K_e} \land w' \sigma \in L(G_e) \land w' \sigma \in \overline{K_e} \implies w' \sigma \in \overline{K_e}).$$

The intuition behind the above definitions can be found in Lin and Wonham (1988a). Finally, let $K$ be a nonempty language satisfying $K = \overline{K} \cap L_m(G)$ that describes the control objective. The existence condition for $\gamma_e$ can then be stated as follows.

**Theorem 1** Assume that $K$ is not empty and $K = \overline{K} \cap L_m(G)$. There exists a nonblocking supervisor $\gamma_e$: $P_e(L(G)) \to 2^{\Sigma}$ such that $L_m(G, \gamma_e) = K$ if and only if $K_e$ is controllable and observable.

**Proof:** Since $K = \overline{K} \cap L_m(G)$, by Lemma 2, $K_e = \overline{K_e} \cap L_m(G_e)$. Therefore, by the results of Lin and Wonham (1988a), $K_e$ is controllable and observable if and only if there exists a nonblocking supervisor $\gamma_e$: $P(L(G_e)) \to 2^{\Sigma_{uc}}$ with $\Sigma_{uc} \subseteq \gamma_e(P(w))$ such that $L_m(G_e, \gamma_e) = K_e$.

Hence the proof reduces to proving that there exists a nonblocking supervisor $\gamma_e$ such that $L_m(G_e, \gamma_e) = K_e$ if and only if there exists a nonblocking supervisor $\gamma_s$ such that $L_m(G, \gamma_s) = K$.

(If):

Suppose there exists a nonblocking supervisor $\gamma_s$ such that $L_m(G, \gamma_s) = K$. Define $\gamma_e$ as follows: For $u = Pw \in P(L(G_e))$ with $w \in L(G_e) = e(L(G))$, let $s \in L(G)$ be the shortest string such that $w \in e(s)$. Let

$$\gamma_e(u) = \gamma_s(P_e(s)) \cup Y \supseteq \Sigma_{uc}.$$

We first show $L(G_e, \gamma_e) = \overline{K_e}$ by an induction on the length of strings.

**Base:** By the definition, $e \in L(G_e, \gamma_e)$. Because $K$ (and hence $K_e$) is nonempty, $e \in \overline{K_e}$. Therefore,

$$e \in L(G_e, \gamma_e) \iff e \in \overline{K_e}.$$ 

**Induction Step:** Assume that for all $w \in (\Sigma \cup Y)^*$ with $|w| \leq n$,

$$w \in L(G_e, \gamma_e) \iff w \in \overline{K_e}.$$ 

Let $s \in L(G)$ be the shortest string such that $w \in e(s)$. Clearly, $T_e(w) = s$. Then for $y \in Y$,

$$wy \in L(G_e, \gamma_e) \iff w \in L(G_e, \gamma_e) \land wy \in L(G_e) \land y \in \gamma_s(Pw) \iff w \in \overline{K_e} \land wy \in L(G_e) \land T_e(wy) = T_e(w) \iff wy \in T_e^{-1}(\overline{K_e} \cap L(G_e)) \iff wy \in T_e^{-1}(\overline{K} \cap L(G_e)) \iff wy \in \overline{K_e}.$$
For $\sigma \in \Sigma$,

$$w\sigma \in L(G_e, \gamma_e)$$

$\Leftrightarrow w \in L(G_e, \gamma_e) \land w\sigma \in L(G_e) \land \sigma \in \gamma_e(P_e(w))$

$\Leftrightarrow w \in L(G_e, \gamma_e) \land w\sigma \in L(G_e) \land \sigma \in \gamma_e(P_e(s))$

$\Leftrightarrow w \in \overline{K_e} \land w\sigma \in L(G_e) \land \sigma \in \gamma_e(P_e(s))$

$\Leftrightarrow w \in \overline{K_e} \land T_e w \in T_eK_e \land w\sigma \in L(G_e) \land \sigma \in \gamma_e(P_e(s))$

$\Leftrightarrow w \in \overline{K_e} \land s \in \overline{K} \land w\sigma \in L(G_e) \land \sigma \in \gamma_e(P_e(s))$

$\Leftrightarrow w \in \overline{K_e} \land s \in L(G, \gamma_e) \land w\sigma \in L(G_e) \land \sigma \in \gamma_e(P_e(s))$

$\Leftrightarrow w \in \overline{K_e} \land s \in L(G, \gamma_e) \land w\sigma \in L(G_e) \land T_e(w\sigma) \in T_e(L(G_e)) \land \sigma \in \gamma_e(P_e(s))$

$\Leftrightarrow w \in \overline{K_e} \land s \in L(G, \gamma_e) \land w\sigma \in L(G_e) \land \sigma \in \gamma_e(P_e(s))$

$\Leftrightarrow w \in \overline{K_e} \land \sigma \in \overline{K}$

$\Leftrightarrow w \in \overline{K_e} \land w\sigma \in L(G_e) \land T_e(w\sigma) \in \overline{K}$

$\Leftrightarrow w \in \overline{K_e} \land w\sigma \in L(G_e) \land w\sigma \in \overline{K}$

Furthermore,

$$L_m(G_e, \gamma_e) = L(G_e, \gamma_e) \cap L_m(G_e) = \overline{K_e} \cap L_m(G_e) = K_e.$$

(ONLY IF):

Suppose there exists a nonblocking supervisor $\gamma_e$ such that $L_m(G_e, \gamma_e) = K_e$. Then define $\gamma_e$ as follows: For $t \in P_e(s) \subseteq P_e(L(G))$ with $s \in L(G)$, let $w$ be the unique element in $e(s) \cap Y(\Sigma Y)^*$. Let

$$\gamma_e(t) = \gamma_e(P_e(w)) \in \overline{S_{uc}}.$$

Then clearly, $(\forall t, t' \in P_e(s)) \gamma_e(t) = \gamma_e(t')$.

We now show $L(G, \gamma_e) = \overline{K}$ by an induction on the length of strings.

Base: Since $K$ is nonempty,

$$\epsilon \in L(G, \gamma_e) \iff \epsilon \in \overline{K}.$$

Induction Step: Assume that for all $s \in \Sigma^*$ with $|s| \leq n$,

$$s \in L(G, \gamma_e) \iff s \in \overline{K}.$$

Then for $\sigma \in \Sigma$, let $w\sigma y$ be the unique element in $e(s\sigma) \cap Y(\Sigma Y)^*$. Clearly $T_e(w) = s$
and we have

\[ \begin{align*}
& s \sigma \in L(G, \gamma_e) \\
\Leftrightarrow &\ s \in L(G, \gamma_e) \land s \sigma \in L(G) \land \sigma \in \gamma_e(P_e(s)) \\
\Leftrightarrow &\ s \in \overline{K} \land s \sigma \in L(G) \land \sigma \in \gamma_e(P_e) \\
\Leftrightarrow &\ s \in \overline{K} \land s \sigma \in L(G) \land T_e(w) \in \overline{K} \land \sigma \in \gamma_e(P_w) \\
\Leftrightarrow &\ s \in \overline{K} \land s \sigma \in L(G) \land T_e(w) \in \overline{K} \land w \sigma \in e(L(G)) \land \sigma \in \gamma_e(P_w) \\
\Leftrightarrow &\ s \in \overline{K} \land s \sigma \in L(G) \land w \in T_e^{-1}\overline{K} \land w \sigma \in L(G_e) \land \sigma \in \gamma_e(P_w) \\
\Leftrightarrow &\ s \in \overline{K} \land s \sigma \in L(G) \land w \in \overline{K}_e \land w \sigma \in L(G_e) \land \sigma \in \gamma_e(P_w) \\
\Leftrightarrow &\ s \in \overline{K} \land s \sigma \in L(G) \land w \sigma \in L(G_e, \gamma_e) \land w \sigma \in L(G_e) \land \sigma \in \gamma_e(P_w) \\
\Leftrightarrow &\ s \in \overline{K} \land s \sigma \in L(G) \land w \sigma \in L(G_e, \gamma_e) \\
\Leftrightarrow &\ s \in \overline{K} \land s \sigma \in L(G) \land w \sigma \in \overline{K}_e \\
\Leftrightarrow &\ s \sigma \in \overline{K}.
\end{align*} \]

and

\[ L_m(G, \gamma_e) = L(G, \gamma_e) \cap L_m(G) = \overline{K} \land L_m(G) = \overline{K}. \]

Since the above proof is constructive, we can synthesize \( \gamma_e \) by first synthesizing \( \gamma_e \) and then letting \( \gamma_e(t) = \gamma_e(P_w) - Y \).

Since state observation will provide more information, we expect that the existence condition derived in this paper will be weaker than the existence condition for a supervisor with event observation only. This is proved in the following three propositions. The first proposition states that controllability is not affected by the state observation.

**PROPOSITION 1** \( K \) is controllable if and only if \( K_e \) is controllable.

**Proof:**

**(IF):**

We prove this part by contradiction. If \( K \) is not controllable, then

\[ (\exists s \in \overline{K})(\exists \sigma \in \Sigma_{ue}) s \sigma \in L(G) \land s \sigma \notin \overline{K}. \]

Let \( w \sigma \gamma \) be the unique element in \( e(s) \cap Y(\Sigma Y)^* \). Then

\[ s \sigma \in L(G) \Rightarrow w \sigma \in e(L(G)) \subseteq L(G_e) \]

\[ T_e(w) = s \in \overline{K} \land w \in e(L(G)) = L(G_e) \Rightarrow w \in \overline{K}_e = T_e^{-1}\overline{K} \land L(G_e) \]

\[ s \sigma = T_e(w \sigma) \notin \overline{K} \Rightarrow w \sigma \notin T_e^{-1}\overline{K} \Rightarrow w \sigma \notin \overline{K}_e. \]

Therefore,

\[ (\exists w \in \overline{K}_e)(\exists \sigma \in \Sigma_{ue}) w \sigma \in L(G_e) \land w \sigma \notin \overline{K}_e. \]

That is, \( K_e \) is not controllable, a contradiction.
(ONLY IF):
We prove this again by contradiction. If \( K_e \) is not controllable, then
\[
(\exists w \in \overline{K}_e)(\exists \sigma \in \Sigma_{ue}) w\sigma \in L(G_e) \land w\sigma \notin \overline{K}_e.
\]
Since \( w\sigma \in L(G_e) = e(L(G)) \), there exists the shortest \( s\sigma \in L(G) \) such that \( w\sigma \in e(s\sigma) \).
Now,
\[
w \in \overline{K}_e = T_e^{-1}\overline{K} \cap L(G_e) \Rightarrow s = T_e(w) \in \overline{K}
\]
\[
w\sigma \notin \overline{K}_e = T_e^{-1}\overline{K} \cap L(G_e) \land w\sigma \in L(G_e) \Rightarrow s\sigma = T_e(w\sigma) \notin \overline{K}.
\]
Therefore,
\[
(\exists s \in \overline{K})(\exists \sigma \in \Sigma_{ue}) s\sigma \in L(G) \land s\sigma \notin \overline{K}.
\]
That is, \( K \) is not controllable, a contradiction.

The next proposition states that observability with event observation only is stronger than observability with both event and state observation.

PROPOSITION 2. If \( K \) is observable then \( K_e \) is observable.

Proof: We prove this proposition by contradiction. If \( K_e \) is not observable, then
\[
(\exists w, w' \in \overline{K}_e)(\exists \sigma \in \Sigma \cup Y) Pw = Pw' \land \sigma_1 \in \overline{K}_e \land \sigma_2 \in L(G_e) \land \sigma_2 \notin \overline{K}_e.
\]
Clearly,
\[
w'\sigma \in L(G_e) \land w' \in \overline{K}_e \land w'\sigma \notin \overline{K}_e \land \overline{K}_e = T_e^{-1}\overline{K} \cap L(G_e) \Rightarrow \sigma \in \Sigma.
\]
Since \( w\sigma, w'\sigma \in L(G_e) = e(L(G)) \), there exist the shortest \( s\sigma, s'\sigma \in L(G) \) such that \( w\sigma \in e(s\sigma) \) and \( w'\sigma \in e(s'\sigma) \). Now by the same argument as in the proof of Proposition 1,
\[
Pw = Pw' \land \sigma_1 \in \overline{K}_e \land w' \in \overline{K}_e \land w'\sigma \in L(G_e) \land w'\sigma \notin \overline{K}_e
\]
\[
\Rightarrow Ps = Ps' \land \sigma \in \overline{K} \land s' \in \overline{K} \land s'\sigma \in L(G) \land s'\sigma \notin \overline{K}.
\]
Therefore,
\[
(\exists s, s' \in \overline{K})(\exists \sigma \in \Sigma) Ps = Ps' \land \sigma \in \overline{K} \land s'\sigma \in L(G) \land s'\sigma \notin \overline{K}.
\]
That is, \( K \) is not observable, a contradiction.

The converse of the above proposition is not true as shown in the following example, because state observation may provide more information.

Example 1. This counterexample shows that \( K_e \) is observable does not imply \( K \) is observable.
Let \( \Sigma = \{a, b, c\} \) with \( \Sigma_o = \{c\} \).
Let \( G \) be the generator in Figure 2 with the transition \( c \) labeled \( \times \) included. The output map \( h \) is also shown in the figure. Clearly,
\[
L(G) = ac + bc.
\]
Let
\[ K = ac + b. \]
That is, the transition \( c \) labeled with \( \times \) in the figure is excluded.

\( K \) is not observable because letting \( s = a, s' = b, \sigma = c \), we have
\[ Ps = Ps' \land s \sigma \in K \land s' \in K \land s' \sigma \in L(G) \land s' \sigma \notin K. \]

However, if we extend \( L(G) \) and \( K \) to \( L(G_e) \) and \( K_e \) as shown in Figure 3, then \( K_e \) is observable. This is because the additional state observation \( y_2 \) and \( y_3 \) help to distinguish two different paths.

Finally, if \( |Y| = 1 \), that is, the (state) output is always the same, then no additional information can be obtained from state observation. Therefore, we expect that observation with or without state observation to be the same.

**Proposition 3.** If \( K_e \) is observable and \( |Y| = 1 \) then \( K \) is observable.

**Proof:** We prove this proposition by contradiction. If \( K \) is not observable, then
\[ (\exists s, s' \in K)(\exists \sigma \in \Sigma) Ps = Ps' \land s \sigma \in K \land s' \sigma \in L(G) \land s' \sigma \notin K. \]
Let \( w \sigma y \) be the unique element in \( e(s \sigma) \cap Y(SU) \) and \( w' \sigma y' \) be the unique element in \( e(s' \sigma) \cap Y(SU) \). Since \( |Y| = 1 \), \( Ps = Ps' \Rightarrow P w = P w' \). By the same argument as in the proof of Proposition 1,
\[ Ps = Ps' \land s \sigma \in K \land s' \in K \land s' \sigma \in L(G) \land s' \sigma \notin K \]
\[ \Rightarrow P w = P w' \land w \sigma \in K_e \land w' \in K_e \land w' \sigma \in L(G_e) \land w' \sigma \notin K_e. \]
Therefore,
\[ (\exists w, w' \in K_e)(\exists \sigma \in \Sigma \cup Y) P w = P w' \land w \sigma \in K_e \land w' \sigma \in L(G_e) \land w' \sigma \notin K_e. \]
That is, $K_e$ is not observable, a contradiction.

Before we conclude our discussion on state observation, we want to add that the main purpose of this discussion is not to suggest that we must convert state observation to event observation. It may well be true that a more efficient algorithm for supervisor synthesis can be obtained without such a conversion. Our discussion does show, however, the following important point: the issue of existence and other theoretical issues can be dealt with entirely within the framework of event observation. The same remark applies to the results in the next section.

3. Event Mask

In Cieslak et al. (1988), a supervisory control problem similar to that of Lin and Wonham (1988a) was studied using event mask instead of event projection. The event mask is defined to be

$$M: \Sigma \rightarrow \Delta \cup \{\epsilon\},$$
where $\Delta$ is the set of (event) output. That is, whenever an event $\sigma$ occurs in $G$, a supervisor can observe $d = M(\sigma)$. If $M(\sigma_1) = M(\sigma_2)$, then $\sigma_1$ and $\sigma_2$ look the same. If $M(\sigma) = \epsilon$, then $\sigma$ cannot be detected. The mapping $M$ is extended to strings and hence $M(L(G))$ is well defined.

A (mask) supervisor is then characterized by a mapping (the subscript $m$ for mask)

$$\gamma_m: M(L(G)) \rightarrow 2^\Sigma,$$

with the usual requirement of $\Sigma_{nc} \subseteq \gamma_m(M(s))$. The languages generated and marked by $\gamma_m$ are defined in the usual way and so is the condition for nonblocking.

The objective is again to synthesize a nonblocking supervisor $\gamma_m$ such that $L_m(G, \gamma_m) = K$ for a given nonempty $K$ satisfying $K = \overline{K} \cap L_m(G)$. To state an existence condition, we define observability with respect to $M$ as

**Definition 3** $K$ is observable (with respect to $L(G)$ and $M$) if

$$\forall s, s' \in \Sigma^* \exists s \in M(s) \Rightarrow \text{consis}(s, s'),$$

where $\text{consis}(s, s')$ is true if and only if

$$\forall \sigma \in \Sigma s\sigma \in \overline{K} \land s'\sigma \in L(G) \land s' \in \overline{K} \Rightarrow s'\sigma \in \overline{K}.$$

In Cieslak et al. (1988), a necessary and sufficient condition is obtained for the existence of a supervisor $\gamma_m$ and it can be restated in terms of the above definition. (The definition of observability was not introduced in Cieslak et al. (1988).)

**Theorem 2** Assume that $K$ is not empty and $K = \overline{K} \cap L_m(G)$. There exists a nonblocking supervisor $\gamma_m: M(L(G)) \rightarrow 2^\Sigma$ such that $L_m(G, \gamma_m) = K$ if and only if $K$ is controllable and observable with respect to $M$.

We will present an alternative way to solve the same problem without using the mask. To this end, we define a substitution operator $f: \Sigma \rightarrow \Sigma \Delta$ as

$$f(\sigma) = \sigma M(\sigma).$$

$f$ can be easily extended to strings. Therefore, $f(L(G))$ is well defined. The generator $G_f$ for the resulting language can be easily obtained from $G$ as illustrated in Figure 4, where each transition $\sigma$ is replaced by $\sigma M(\sigma)$. For such a $G_f$, $L(G_f) = \overline{f(L(G))}$ and $L_m(G_f) = f(L_m(G))$. Similar to the case of state observation, we can then extend the desired language $K$ as

$$K_f = T_f^{-1} \overline{K} \cap L_m(G_f),$$

where $T_f: (\Sigma \cup \Delta)^* \rightarrow \Sigma^*$ is the projection.

Some properties of $K_f$ are stated in the following lemma.

**Lemma 3** Assume that $L(G)$ is not empty and $K = \overline{K} \cap L_m(G)$. Then

(i) $T_f L(G_f) = L(G)$
Figure 4.

(ii) \( \overline{K_f} = T_{f^{-1}} \overline{K} \cap L(G_f) \)

(iii) \( K_f = \overline{K_f} \cap L_m(G_f) \)

(iv) \( T_f \overline{K_f} = \overline{K} \).

**Proof:**

(i) Clearly \( T_f(f(s)) = \overline{s} \) for all \( s \in L(G) \). Therefore,
\[
T_f(L(G_f)) = T_f(f(L(G))) = \overline{L(G)} = L(G).
\]

(ii) \((\subseteq)\):
\[
\overline{K_f} = \frac{T_{f^{-1}} \overline{K} \cap L_m(G_f)}{\subseteq T_{f^{-1}} \overline{K} \cap L_m(G_f)} = T_{f^{-1}} \overline{K} \cap L(G_f).
\]

\((\supseteq)\):
\[
w \in T_{f^{-1}} \overline{K} \cap L(G_f) \\
\Rightarrow w \in L(G_f) \wedge T_f w \in \overline{K} \\
\Rightarrow w \in f(L(G)) \wedge T_f w \in \overline{K} \\
\Rightarrow (\exists s \in L(G)) w \in \overline{f(s)} \wedge T_f w = s \in \overline{K} \\
\Rightarrow (\exists s \in L(G)) w \in \overline{f(s)} \wedge (\exists t \in \Sigma^*) st \in K \\
\Rightarrow (\exists s \in L(G)) w \in \overline{f(s)} \wedge (\exists t \in \Sigma^*) st \in \overline{K} \cap L_m(G) \\
\Rightarrow (\exists s \in L(G))(\exists t \in \Sigma^*)(\exists v \in (\Sigma \cup \Delta)^*) w v = f(st) \wedge st \in L_m(G) \wedge st \in \overline{K} \\
\Rightarrow (\exists s \in L(G))(\exists t \in \Sigma^*)(\exists v \in (\Sigma \cup \Delta)^*) w v = f(st) \\
\wedge uv \in f(L_m(G)) \wedge st \in \overline{K} \\
\Rightarrow (\exists s \in L(G))(\exists t \in \Sigma^*)(\exists v \in (\Sigma \cup \Delta)^*) w v = f(st) \\
\wedge uv \in L_m(G_f) \wedge T_f(wv) \in \overline{K} \\
\Rightarrow (\exists v \in (\Sigma \cup \Delta)^*) wv \in T_{f^{-1}} \overline{K} \cap L_m(G_f) \\
\Rightarrow w \in K_f.
\]
(iii)
\[
\overline{K_f} \cap L_m(G_f) = T_f^{-1} \overline{K} \cap L(G_f) \cap L_m(G_f) \\
= T_f^{-1} \overline{K} \cap L_m(G_f) \\
= K_f.
\]

(iv) (⊆):
\[
T_f \overline{K_f} = T_f(T_f^{-1} \overline{K} \cap L(G_f)) \\
\subseteq T_f(T_f^{-1} \overline{K}) \cap T_f L(G_f) \\
= \overline{K} \cap L(G) \\
= \overline{K}.
\]

(⊇):
\[
s \in \overline{K} \\
s' \in \overline{K} \land s' \in L(G) \\
s \in \overline{K} \land f(s) \in f(L(G)) \\
s \in \overline{K} \land f(s) \in L(G_f) \\
(\exists w = f(s)) T_f w \in \overline{K} \land w \in L(G_f) \\
(\exists w = f(s)) T_f w \in T_f^{-1} \overline{K} \cap L(G_f) \\
(\exists w = f(s)) w \in \overline{K_f} \\
s \in T_f \overline{K_f}.
\]

Now an alternative way to state the existence condition is as follows (we will use \( P \) to denote the projection from \((\Sigma \cup \Delta)^*\) to \( \Delta^* \) in the rest of this section).

**Theorem 3** Assume that \( K \) is not empty and \( K = \overline{K} \cap L_m(G) \). There exists a nonblocking supervisor \( \gamma_m: M(L(G)) \to 2^\Sigma \) such that \( L_m(G, \gamma_m) = K \) if and only if \( K_f \) is controllable and observable with respect to \( P \).

**Proof:** Since \( K = \overline{K} \cap L_m(G) \), by Lemma 3, \( K_f = \overline{K_f} \cap L_m(G_f) \). Therefore, by the results of Lin and Wonham (1988a), \( K_f \) is controllable and observable with respect to \( P \) if and only if there exists a nonblocking supervisor \( \gamma_f: P(L(G_f)) \to 2^{\Sigma \cup \Delta} \) with \( \Sigma_{uc} \subseteq \gamma_f(P(w)) \) such that \( L_m(G_f, \gamma_f) = K_f \).

Hence the proof reduces to proving that there exists a nonblocking supervisor \( \gamma_f \) such that \( L_m(G_f, \gamma_f) = K_f \) if and only if there exists a nonblocking supervisor \( \gamma_m \) such that \( L_m(G, \gamma_m) = K \).

(IFF):

Suppose there exists a nonblocking supervisor \( \gamma_m \) such that \( L_m(G, \gamma_m) = K \). Define \( \gamma_f \) as follows: For \( u = Pw \in P(L(G_f)) \) with \( w \in L(G_f) = f(L(G)) \), let \( s \in L(G) \) be the shortest string such that \( w \in f(s) \). Let
\[
\gamma_f(u) = \gamma_m(Ms) \cup \Delta \supseteq \Sigma_{uc}.
\]

We first show \( L(G_f, \gamma_f) = \overline{K_f} \) by an induction on the length of strings.
Base: By the definition, $\epsilon \in L(G_f, \gamma_f)$. Because $K$ (and hence $K_f$) is nonempty, $\epsilon \in \overline{K_f}$. Therefore,

$$\epsilon \in L(G_f, \gamma_f) \iff \epsilon \in \overline{K_f}.$$ 

Induction Step: Assume that for all $w \in (\Sigma \cup \Delta)^*$ with $|w| \leq n$,

$$w \in L(G_f, \gamma_f) \iff w \in \overline{K_f}.$$ 

Let $s \in L(G)$ be the shortest string such that $w \in \overline{f(s)}$. Clearly $T_f(w) = s$. Then for $d \in \Delta$,

$$wd \in L(G_f, \gamma_f)$$

$$\iff w \in L(G_f, \gamma_f) \land wd \in L(G_f) \land d \in \gamma_f(Pw)$$

$$\iff w \in \overline{K_f} \land wd \in L(G_f) \land T_f(wd) = T_f(w)$$

$$\iff wd \in T_f^{-1}K \cap L(G_f)$$

$$\iff wd \in \overline{K_f} \cap L(G_f)$$

For $\sigma \in \Sigma$,

$$w\sigma \in L(G_f, \gamma_f)$$

$$\iff w \in L(G_f, \gamma_f) \land w\sigma \in L(G_f) \land \sigma \in \gamma_f(Pw)$$

$$\iff w \in L(G_f, \gamma_f) \land w\sigma \in L(G_f) \land \sigma \in \gamma_m(Ms)$$

$$\iff w \in \overline{K_f} \land w\sigma \in L(G_f) \land \sigma \in \gamma_m(Ms)$$

$$\iff w \in \overline{K_f} \land s \in L(G, \gamma_m) \land w\sigma \in L(G_f) \land \sigma \in \gamma_m(Ms)$$

$$\iff w \in \overline{K_f} \land s \in L(G, \gamma_m) \land w\sigma \in L(G_f) \land \sigma \in \gamma_m(Ms)$$

$$\iff w \in \overline{K_f} \land s \in L(G, \gamma_m) \land w\sigma \in L(G_f) \land \sigma \in \gamma_m(Ms)$$

$$\iff w \in \overline{K_f} \land w\sigma \in L(G_f) \land \sigma \in \overline{K}$$

$$\iff w \in \overline{K_f} \land w\sigma \in L(G_f) \land \sigma \in \overline{K}$$

$$\iff w \in \overline{K_f} \land w\sigma \in L(G_f) \land \sigma \in T_f^{-1}\overline{K}$$

$$\iff w\sigma \in \overline{K_f}.$$ 

Finally,

$$L_m(G_f, \gamma_f) = L(G_f, \gamma_f) \cap L_m(G_f) = \overline{K_f} \cap L_m(G_f) = K_f.$$ 

(ONLY IF):

Suppose there exists a nonblocking supervisor $\gamma_f$ such that $L_m(G_f, \gamma_f) = K_f$. Then define $\gamma_m$ as follows: For $t = Ms \in M(L(G))$ with $s \in L(G)$, let $w = f(s)$ and

$$\gamma_m(t) = \gamma_f(Pw) - \Delta \supseteq \Sigma_{uc}.$$ 

We now show $L(G, \gamma_m) = \overline{K}$ by an induction on the length of strings.
Base: Since $K$ is nonempty,
\[ \epsilon \in L(G, \gamma_m) \Leftrightarrow \epsilon \in K. \]

Induction Step: Assume that for all $s \in \Sigma^*$ with $|s| \leq n$,
\[ s \in L(G, \gamma_m) \Leftrightarrow s \in K. \]

Then for $\sigma \in \Sigma$, let $w\sigma d = f(s\sigma)$. Clearly, $T_f(w) = s$ and
\[ s\sigma \in L(G, \gamma_m) \]
\[ \Leftrightarrow s \in L(G, \gamma_m) \land s\sigma \in L(G) \land \sigma \in \gamma_m(Ms) \]
\[ \Leftrightarrow s \in K \land s\sigma \in L(G) \land \sigma \in \gamma_m(Ms) \]
\[ \Leftrightarrow s \in K \land s\sigma \in L(G) \land \sigma \in \gamma_f(Ps) \]
\[ \Leftrightarrow s \in \overline{K} \land s\sigma \in L(G) \land \sigma \in \gamma_f(Ps) \]
\[ \Leftrightarrow s \in \overline{K} \land s\sigma \in L(G) \land T_f(w) \in K \land \sigma \in \gamma_f(Pw) \]
\[ \Leftrightarrow s \in \overline{K} \land s\sigma \in L(G) \land T_f(w) \in K \land w\sigma \in f(L(G)) \land \sigma \in \gamma_f(Pw) \]
\[ \Leftrightarrow s \in \overline{K} \land s\sigma \in L(G) \land T_f(w) \in K \land w\sigma \in L(G_f) \land \sigma \in \gamma_f(Pw) \]
\[ \Leftrightarrow s \in \overline{K} \land s\sigma \in L(G) \land w \in T_f^{-1}(K \land w\sigma \in L(G_f) \land \sigma \in \gamma_f(Pw) \]
\[ \Leftrightarrow s \in \overline{K} \land s\sigma \in L(G) \land w \in L(G_f, \gamma_f) \land w\sigma \in L(G_f) \land \sigma \in \gamma_f(Pw) \]
\[ \Leftrightarrow s \in \overline{K} \land s\sigma \in L(G) \land w\sigma \in K \]
\[ \Leftrightarrow s \in \overline{K} \land s\sigma \in L(G) \land \sigma \in \overline{K} \]
\[ \Leftrightarrow s\sigma \in \overline{K}. \]

and
\[ L_m(G, \gamma_m) = L(G, \gamma_m) \cap L_{m'}(G) = \overline{K} \cap L_m(G) = K. \]

Since the above proof is constructive, we can synthesize $\gamma_m$ by first synthesizing $\gamma_f$ and then letting $\gamma_m(t) = \gamma_f(Pw) - \Delta$.

From Theorems 2 and 3, we have the following conclusion: $K$ is controllable and observable with respect to $M$ if and only if $K_f$ is controllable and observable with respect to $P$. If we let all events be observable, then the following corollary can be proved.

COROLLARY 1 $K$ is controllable if and only if $K_f$ is controllable.

Similarly, by letting all events be controllable, we have the following corollary.

COROLLARY 2 $K$ is observable with respect to $M$ if and only if $K_f$ is observable with respect to $P$.

The results in this section show that in terms of expressive power, event mask is no more powerful than event projection, because an event mask problem can be made equivalent to an event projection problem.

4. Nondeterministic Systems

Observability is also very useful in studying nondeterminism as we will show in this section. Currently, there is a renewed interest in nondeterministic systems (Heymann 1990, Heymann and Meyer 1991, Kumar and Shayman 1993, Overkamp 1994, Shayman and Kumar
1995), because in some applications, deterministic models are not adequate. For example, consider a discrete event system modeled by the deterministic automaton in Figure 5. If the events \( c \) and \( d \) represent the internal underlying dynamics undetectable from the outside, then from the outside, the system is modeled by the nondeterministic automaton in Figure 5. This system cannot be accurately described by the language generated by the system \( L(G) = \bar{a}b \), because it does not account for the fact that the system may not be able to execute any event after \( a \). Hence we cannot simply convert the nondeterministic automaton in Figure 5 into a deterministic one and proceed as usual.

We will show that the above problem of nondeterminism is closely related to observability. To do this, we need to first answer the following question: What is nondeterminism? The answer is not as simple as one may think. For example, the nondeterministic automaton in Figure 6 does not represent a nondeterministic system because it can be easily converted to a deterministic automaton as illustrated in Figure 6. In fact, there are several competing definitions of nondeterminism, most notably the definition in Hoare (1985) (see also Heymann (1990), Heymann and Meyer (1991), Heymann and Lin (1996), Kumar and Shayman (1993), Overkamp (1994), Shayman and Kumar (1995)). The definition to be introduced in this paper is weaker than that in Hoare (1985) (in other words, a system may be deterministic by the definition of Hoare (1985) but nondeterministic by our definition). The following example illustrates the motivation of introducing our definition.

**Example 2** Consider the system in Figure 7(a). The events \( a, b, \) and \( c \) are defined as follows:

- \( a \): stop at an intersection;
- \( b \): wait for the traffic light to turn green; and
- \( c \): cross the intersection.

If the event \( b \) is unobservable, then when the event \( c \) can occur will be uncertain forever. (One can imagine that there is another driver on another direction who needs to decide when to cross the intersection.) The resulting system by internalizing the event \( b \) is shown
in Figure 7(b). This system is deterministic according to the definition in Hoare (1985). However, it is obvious that there is a certain degree of nondeterminism in the system that should not be taken lightly. (Especially if someone’s life is in question).

To define nondeterminism that captures the above phenomenon, we first define, for a system $G_o$, the refusals of $G_o$ after a string $s \in L(G_o)$, $refusals(G_o, s)$, as the set of all $X \subseteq \Sigma$ such that it is possible that $G_o$ cannot execute any event in $X$ after $s^1$. If $G_o$ is represented by a nondeterministic automaton

$$G_o = (\Sigma \cup \{\epsilon\}, Q, \delta_o, q_o)$$

with $\delta_o: (\Sigma \cup \{\epsilon\}) \times Q \rightarrow 2^Q$ a nondeterministic transition function, then

$$refusals(G_o, s) = \{X \subseteq \Sigma: (\exists q \in \delta_o(s, q_o))(\forall \sigma \in X)\delta_o(\sigma, q) = \emptyset\},$$

where $\delta_o(s, q_o)$ is the set of states reachable from $q_o$ via $s$ with possible $\epsilon$-transitions. We now give the following definition of nondeterminism.
Definition 4 \( G_o \) is deterministic if
\[
(\forall X \subseteq \Sigma)(\forall s \in L(G_o))(X \in \text{refusals}(G_o, s) \Leftrightarrow X \cap (L(G_o)/s) = \emptyset),
\]
where \( L(G_o)/s = \{t \in \Sigma^*: st \in L(G_o)\} \).

This definition captures the nondeterminism discussed in Example 2: Let \( G_o \) be the nondeterministic automaton in Figure 7(b), then
\[
\{c\} \in \text{refusal}(G_o, a),
\]
but
\[
\{c\} \cap (L(G_o)/a) = \{c\} \neq \emptyset.
\]
So by our definition, \( G_o \) is nondeterministic, but it is deterministic by the definition in Hoare (1985).

We would like to show that nondeterminism is closely related to observability. To this end, we denote the set of internal events by \( \Sigma_{in} \) and let \( P_{in}: (\Sigma \cup \Sigma_{in})^* \rightarrow \Sigma^* \) be the projection. Consider a system whose dynamics including the internal dynamics is described by a deterministic automaton
\[
G_{in} = (\Sigma \cup \Sigma_{in}, Q, \delta_{in}, q_0, Q_m),
\]
where \( \delta_{in}: (\Sigma \cup \Sigma_{in}) \times Q \rightarrow Q \) is the (deterministic) transition function.

Now we extend our definition of observability to \( G_{in} \) as follows.

Definition 5 \( G_{in} \) is observable (with respect to \( (\Sigma \cup \Sigma_{in})^* \) and \( P_{in} \)) if
\[
(\forall w, w' \in (\Sigma \cup \Sigma_{in})^*)(P_{in}w = P_{in}w' \Rightarrow \text{consis}(w, w')),
\]
where \( \text{consis}(w, w') \) is true if and only if
\[
(\forall \sigma \in \Sigma)(w \sigma \in L(G_{in}) \land w' \in L(G_{in}) \Rightarrow w' \sigma \in L(G_{in})).
\]

In other words, in defining observability of \( G_{in} \), we take \( L(G_{in}) \) as “legal” behavior and \( (\Sigma \cup \Sigma_{in})^* \) as “physically possible” behavior. Let \( G_o \) be the nondeterministic automaton obtained by replacing transitions in \( G_{in} \) labeled by \( \sigma \in \Sigma_{in} \) with \( \varepsilon \). Then we can prove the following result.

Theorem 4 \( G_o \) is deterministic if and only if \( G_{in} \) is observable.

Proof: \( G_o \) is deterministic if and only if, for all \( X \subseteq \Sigma, \)
\[
(\forall s \in L(G_o))(X \in \text{refusals}(G_o, s) \Leftrightarrow X \cap (L(G_o)/s) = \emptyset).
\]

By the definition of refusals,
\[
X \in \text{refusals}(G_o, s)
\Leftrightarrow (\exists q \in \delta(s, q_0))(\forall \sigma \in X)\delta_o(\sigma, q) = \emptyset
\Leftrightarrow (\exists t \in L(G_{in}))(P_{in}t = s \land X \cap (L(G_{in})/t) = \emptyset).
\]

(1)
In other words, \( X \in \text{refusals}(G_o, s) \) iff there is a string \( t \) in \( L(G_{in}) \) that looks like \( s \) after projection and no event in \( X \) is possible after \( t \).

On the other hand,

\[
X \cap (L(G_o)/s) = \emptyset \\
\iff (\forall t \in L(G_{in}))(P_{int} = s \Rightarrow X \cap (L(G_{in})/t) = \emptyset).
\]  

(2)

That is, \( X \cap (L(G_o)/s) = \emptyset \) iff no event in \( X \) is possible after all strings in \( L(G_{in}) \) that look like \( s \) after projection.

From (1) and (2), it is clear that the left implication in the definition of determinism is automatically satisfied. Hence, \( G_o \) is deterministic if and only if, for all \( X \subseteq \Sigma \),

\[
(\forall s \in L(G_o))(X \in \text{refusals}(G_o, s) \Rightarrow X \cap (L(G_o)/s) = \emptyset).
\]

By (1) and (2) this is equivalent to

\[
(\forall s \in L(G_o))((\exists t \in L(G_{in}))(P_{int} \neq s \land X \cap (L(G_{in})/t) = \emptyset) \Rightarrow (\forall t \in L(G_{in}))(P_{int} = s \Rightarrow X \cap (L(G_{in})/t) = \emptyset)).
\]

Since the above is true for all \( X \subseteq \Sigma \), it is equivalent to

\[
(\forall \sigma \in \Sigma)(\forall s \in L(G_o))((\exists t \in L(G_{in}))(P_{int} \neq s \land \sigma \notin (L(G_{in})/t)) \Rightarrow (\forall t \in L(G_{in}))(P_{int} = s \Rightarrow \sigma \notin (L(G_{in})/t))
\]

(3)

\[
\iff (\forall \sigma \in \Sigma)(\forall t \in L(G_{in}))(\sigma \in (L(G_{in})/t)
\]

\[
\Rightarrow (\forall t \in L(G_{in}))(P_{int} = P_{int} \land \sigma \notin (L(G_{in})/t))
\]

(4)

\[
\iff (\forall \sigma \in \Sigma)(\forall t \in L(G_{in}))(\sigma \in (L(G_{in})/t)
\]

\[
\Rightarrow (\forall t \in L(G_{in}))(P_{int} = P_{int} \Rightarrow \sigma \in (L(G_{in})/t))
\]

(5)

On the other hand, \( G_{in} \) is observable if and only if

\[
(\forall w, w' \in (\Sigma \cup \Sigma_{in})^*) (P_{in}w = P_{in}w' \Rightarrow \text{consis}(w, w'))
\]

(6)

\[
\iff (\forall w, w' \in (\Sigma \cup \Sigma_{in})^*) (P_{in}w = P_{in}w' \\
\Rightarrow (\forall \sigma \in \Sigma)(w\sigma \in L(G_{in}) \land w' \in L(G_{in}) \Rightarrow w'\sigma \in L(G_{in})))
\]

\[
\iff (\forall \sigma \in \Sigma)(\forall w, w' \in (\Sigma \cup \Sigma_{in})^*) (P_{in}w = P_{in}w' \\
\Rightarrow (w\sigma \in L(G_{in}) \land w' \in L(G_{in}) \Rightarrow w'\sigma \in L(G_{in})))
\]

\[
(\forall \sigma \in \Sigma)(\forall w, w' \in L(G_{in}))(P_{in}w = P_{in}w' \Rightarrow (w\sigma \in L(G_{in}) \Rightarrow w'\sigma \in L(G_{in})))
\]

which is equivalent to (3). Therefore, \( G_o \) is deterministic if and only if \( G_{in} \) is observable.
The above theorem shows an intrinsic relationship between nondeterminism and observability. Based on this intuition, a theory for supervisory control of nondeterministic systems is developed in Heymann and Lin (1996).

5. Conclusion

We have shown three new applications of observability theory to (1) state observation, (2) event mask, and (3) nondeterminism. These results are of theoretical importance because they demonstrate that the essence of the three problems are all captured by the original definition of observability.

Acknowledgment

This research is supported in part by the National Science Foundation under grant ECS-9315344. We would like to acknowledge the fruitful discussions with Ard Overkamp that resulted in the modification of Section 4. We also like to thank the anonymous reviewers and the associate editor for their very helpful comments.

Notes

1. This definition of refusal is different from that of Hoare (1985).

References


