Generalised projections in finite state automata and decidability of state determinacy

Ishanu Chattopadhyay and Asok Ray*

Pennsylvania State University, University Park, USA

(Received 16 October 2007; final version received 2 January 2008)

Loss of sensors and communication links may lead to incomplete observation at the supervisory level of discrete event systems (DES). Under these circumstances, an event may conceivably be observable at one state and unobservable at another state and the observability may become dependent on the history of event occurrences. This paper presents a framework for analysis of generalised projection maps in DES, including the maps that introduce possibly unbounded memory.

Keywords: formal languages; partial observation; discrete event systems

1. Introduction

Incomplete observation in discrete event systems (DES) may introduce (possibly unbounded) additional memory in the observed plant model and thus complicate the problem of supervisory decision and control. Specifically, loss of sensors and communication links may lead to incomplete observation at the supervisory level, where an event may conceivably be observable at one state and unobservable at another state and the observability may become dependent on the history of event occurrences.

Natural projection maps have played a central role in the study of decentralised control and incomplete observation in DES (Lin and Wonham 1988a,b). This paper considers a generalisation of natural projections which we refer to as unobservability maps. Specifically, we investigate situations where the observability of a particular transition from a given state depends on (possibly unbounded) event history. Wong (1988) has shown the state size of a projected plant to be a possibly exponential function of that of the original system. As an extension of this result, this paper shows that the observed plant language may become non-regular for generalised projections, i.e., fail to have any finite state description. Two central concepts, namely, projective Nerode equivalence and p-minimal plant realisation, are introduced for an unobservability map p. The notion of $\Sigma$-normal representations is introduced for p-minimal realisations of unobservability. It is shown that such representations allow finite language-theoretic (but not necessarily finite state) descriptions for complex unobservability situations.

Following a formalisation of the notion of generalised unobservability in DES, this paper investigates the state determinacy problem (Chattopadhyay and Ray 2007b). Given an observed event trace and an unobservability map, the problem is to compute the set of states that the plant currently can possibly be in. The above problem is conceptually related to diagnosability of DES introduced by Sampath et al. (1995). Given an observed event trace, DES diagnosis is the problem of determining if an unobservable event has occurred. Assuming state-independent regular unobservability, as defined in §4, and two important restrictions (i.e., liveness and absence of unobservable cycles) on the underlying plant dynamics, Sampath et al. have shown that, for diagnosable systems, the complexity of diagnoser construction and its subsequent execution is exponential in the number of states and doubly exponential in the number of “failure types”. Jiang et al. (2003 a,b) have extended this notion to state-based $k$-diagnosability, where the restriction on unobservable cycles is relaxed and polynomial-time algorithms for diagnosability checking is reported. State determinacy differs from diagnosis in the sense that, given the unobservability map, a state estimate must be obtained immediately after each observed transition, irrespective of system diagnosability. For a finite state automaton, this paper shows that although the state determinacy problem can be solved in polynomial time for regular unobservability maps, it is undecidable for more complex situations.

The above analysis is important in cases where one cannot “arrange” for some specific property (e.g., the
observer property) by suitably modifying the hierarchy (Wong and Wonham 2004), or by an optimal sensor selection (Jiang et al. 2003b). In such a situation, one must work with the given unobservability map that represents operations of the physical plant, which may turn out to have unbounded memory. Thus even with a finite state underlying plant, the observed behaviour may be non-regular. Since any realistic and practically implementable supervisor must take control decisions based on observation of transpired events, supervisor design must be based on an observed non-regular language.

The paper is organised in seven sections including the present one. Section 2 provides preliminary concepts and notations as the background material. Section 3 introduces unobservability maps and their minimal realisation to bring in the associated concept of projective Nerode equivalence. Section 4 deals with $\Sigma$-normal representations for $p$-minimal realisations of unobservability. Section 5 introduces the State Determinacy problem and derives formal conditions that are necessary for the former to be decidable. Section 6 presents a simple example and the paper is summarised in Section 7 along with recommendations for future work.

2. Preliminary concepts and notations

This section delineates pertinent notations and preliminary concepts following the standard literature on discrete event systems (Ramadge and Wonham 1987; Ray 2005; Chattopadhyay and Ray 2007a) and formal language theory (Hopcroft et al. 2001).

2.1 Notation

Logical AND and OR operations are indicated by $\land$ and $\lor$, respectively; if $G$ is a state transition system with $Q$ as the set of states, then $G\downarrow$ and $G\uparrow$ imply finite and (countably) infinite cardiality of the state set $Q$, respectively; $\text{REG}$ denotes the set of regular languages; $|\omega|$ is used interchangeably with the expression “such that”; $|\omega|$ denotes the (non-negative integer) length of the event string $\omega$; and $\mathbb{N}$ is the set of natural numbers. If $N$ is an equivalence relation, then $xN\ U\ y$ implies that $x$ and $y$ belong to the same equivalence class of $N$, while $x|N\ U\ y$ implies that they belong to distinct equivalence classes.

A trim (i.e., accessible and co-accessible) deterministic finite-state automaton (DFSA) $G_i=(Q, \Sigma, \delta, q_i, O_m)$ represents the discrete-event dynamics of a physical plant, where $Q=\{q_k:k \in \mathcal{I}_Q\}$ is the set of states and $\mathcal{I}_Q=\{1, 2, \ldots, n\}$ is the index set of states; the automaton starts with the initial state $q_i$; the alphabet of events is $\Sigma=\{\sigma_k:k \in \mathcal{I}_x\}$, having $\Sigma \cap \mathcal{I}_Q=\emptyset$ and $\mathcal{I}_Q=\{1, 2, \ldots, n\}$ is the set of index of events; $\delta: \Sigma \times \Sigma \rightarrow Q$ is the (possibly partial) function of state transitions; and $O_m=\{q_{m_1}, q_{m_2}, \ldots, q_{m_l}\} \subseteq Q$ is the set of marked (i.e., accepting) states with $q_{m_2}=q_j$ for some $j \in \mathcal{I}_Q$. Let $\Sigma^*$ be the Kleene closure of $\Sigma$, i.e., the set of all finite-length strings made of the events belonging to $\Sigma$ as well as the empty string $\epsilon$ that is viewed as the identity of the monoid $\Sigma^*$ under the operation of string concatenation, i.e., $\epsilon s=s=se$. The extension $\delta^*: \Sigma^* \times \Sigma^* \rightarrow Q$ is defined recursively in the usual sense (Ramadge and Wonham 1987; Hopcroft et al. 2001). The prefix closure of a language $L$ is denoted by $\mathcal{L}$.

2.2 Partial and total automata models

Partial DFSA refers to models $G_i=(Q, \Sigma, \delta, q_i, O_m)$ for which the transition function $\delta$ is partial as opposed to complete models for which $\delta$ is a total function. In the paradigm of discrete event control, partial automata are often used to model plant dynamics (Ramadge and Wonham 1987). It is always possible to complete a partial transition function by adding a non-marked dump state $q_d$ and allocating the missing transitions to $q_d$. In doing so, however, the model no longer remains trim and some key results in supervisory control theory (Ramadge and Wonham 1987) require the aforementioned property. The technical problem in dealing with partial models is whether the results of classical automata theory (e.g. existence of minimal representations) (Hopcroft et al. 2001) carry over to this case. In this regard, two possible approaches are:

1. completion of the automaton by introducing a single dump state;
2. extension of the notion of acceptance of strings by finite state automata, by defining a string to be accepted if and only if the final state is accepting and if the automaton does not block, i.e., can process the entire string.

In sequel, the second approach is adopted to preserve the trim property of the given partial DFSA model.

2.3 Accepted language and generated language

Definition 1: The (regular) language accepted by a DFSA $G_i=(Q, \Sigma, \delta, q_i, O_m)$ is defined to be $L_m(G_i)=s \in \Sigma^*|\delta^*(q_i, s) \in O_m$. $L_m(G_i)$ is also referred to as the marked language.

Definition 2: The (regular) language generated by a DFSA $G_i=(Q, \Sigma, \delta, q_i, O_m)$ is defined to be the set of strings $\{s \in \Sigma^*|\delta^*(q_i, s) \in Q\}$.
It follows that, for a trim DFSA, the generated language is the prefix closure of the marked (accepted) language. In the sequel, only trim models are considered and hence the generated plant language of a DFSA model $G_i=(Q, \Sigma, \delta, q_i, Q_m)$ is denoted as $L_m(G_i)$. Note that if the transition function $\delta$ is total, then $L_m(G_i) = \Sigma^*$. Also, if all states are marked, i.e., if $Q = Q_m$ then $L_m(G_i) = L_m(G_i)$.

**Definition 3:** The language $L(q_i, q_j) \subseteq L_m(G_i)$ is defined as: $L(q_i, q_j) = \{ s \in \Sigma^* | \delta^*(q_i, s) = q_j \}$.

3. Minimal realisation of unobservability maps

In Ramadge and Wonham’s original treatment of unobservability (Ramadge and Wonham 1987), the symbol alphabet $\Sigma$ is partitioned as $\Sigma_o \cup \Sigma_{uo}$, where events in $\Sigma_o$ are observable from all states at which they are defined while the events in $\Sigma_{uo}$ are always unobservable. This is equivalent to considering a natural projection of a given DES (Wong 1998).

**Definition 4:** Given a DFSA $G_i$ with the (regular) marked language $L_m(G_i)$ a generalised projection $p_i : L_m(G_i) \rightarrow \Sigma^*$ is a mapping that satisfies the following conditions:

1. $p(\epsilon) = \epsilon$;
2. $p(\sigma) \in \{ \sigma, \epsilon \} \forall \sigma \in \Sigma$;
3. $p(h\sigma) \in \{ p(h)\sigma, p(h) \} \forall \sigma \in \Sigma, h \in \Sigma^*$.

In the sequel, such maps are referred to as unobservability maps. If the initial state is understood in the context, the subscript $i$ in $p_i$ may be dropped.

**Definition 4** does not imply that $p(h_1 h_2) = p(h_1) p(h_2)$, i.e., $p$ is not restricted to be a morphism as in the case of natural projections (Wong 1998; Lin and Wonham 1988a,b). Unobservability maps, in fact, belong to a restricted subclass of general causal reporter maps (Wong 1998). In contrast to a causal reporter map that takes a string over a given alphabet $\Sigma$ to another string over a possibly distinct alphabet $\Gamma$, both the image and pre-image are strings over the same alphabet for unobservability maps. It follows that unobservability maps are prefix-preserving (Wong 1998), i.e., if $s$ is a prefix of $s'$, then $p(s)$ is a prefix of $p(s')$ for an unobservability map $p$.

**Definition 5:** For a DFSA $G_i$ with generated language $L_m(G_i)$ and a given unobservability map $p_i$, the observed language $\Theta_p$ is defined as $\{ w \in \Sigma^* | \exists s \in L_m(G_i) \text{ such that } p_i(s) = w \}$.

**Remark 1:** Since unobservability maps are prefix-preserving and generated languages are prefix closed, the observed language $\Theta_p$ is prefix closed.

The necessity of considering unobservability maps, which are not morphisms, is illustrated in Figure 1 and 2. The unobservability map in Figure 1 is completely described by marking all transitions labelled by $\beta$ in the automaton as unobservable. The unobservability map in this case is a morphism and is an example of state-independent unobservability or a natural projection according to Wonham (2001). In Figure 2, the observability of an event is dependent on the state from which it is generated. Although a partitioning of the event alphabet into observable and unobservable parts is not feasible in this case, a finite description of the unobservability map is still possible as shown in Figure 2. This unobservability map is definitely not a morphism, but a simple 2-colouring or marking of edges of the graph to represent the finite-state automaton is sufficient to completely describe the unobservability. Such a description is called a marked realisation of the unobservability map in the sequel. The map $p_i$ can be completely specified by enumerating the image of each string in $L_m(G_i)$, which is a prefix-closed language. However, this approach may not to able to provide ample insight into the structure of the unobservability map. Furthermore, if the language $L_m(G_i)$ is infinite, such an approach may not work. In this context, the binary operation of prefix subtraction is introduced, which is required to define induced unobservability maps.

**Definition 6:** Let $s_1, s_2 \in \Sigma^*$ and let $\lambda$ be the longest common prefix of the event strings $s_1$ and $s_2$, i.e., $s_1 = \lambda \omega_1$ and $s_2 = \lambda \omega_2$ with $\omega_1, \omega_2 \in \Sigma^*$ such that $\omega_1$ and
Thus, under state-dependent unobservability, an event $q_k$ is unobservable at some state $q_k$ if an event $q_k$ is unobservable in the “marked realisation” for one possible path reaching state $q_k$, then it is unobservable for all paths that reach the same state $q_k$.

The term “marked” in “marked realisation” denotes that transitions can be unambiguously denoted as either observable or unobservable. The formalisation of this concept requires the notion of a (possibly infinite state) deterministic transition system (Leeuwen 1990).

Definition 10: A deterministic labelled transition system $\mathcal{T}$ is a tuple $(\bar{Q}, \Sigma, \delta, \bar{q}_i)$ such that

1. $\bar{Q}$ is the (possibly infinite) set of states,
2. $\Sigma$ is the alphabet,
3. $\bar{q}_i$ is the initial state,
4. $\delta \subseteq Q \times \Sigma \times \bar{Q}$ is a ternary relation such that

$$\forall \bar{q}_j \in \bar{Q} \forall \sigma \in \bar{\Sigma} ((\exists \bar{q}_k \in \bar{Q} \text{ such that } (\bar{q}_j, \sigma, \bar{q}_k) \in \delta) \Rightarrow \exists \bar{q}_i \in \bar{Q} \setminus \bar{q}_k (\bar{q}_j, \sigma, \bar{q}_i) \notin \delta).$$

Property 3 ensures that $\mathcal{T}$ is deterministic.

Definition 11 (marked realisation): Let $p_i$ be an unobservable map for a DFSA $G_i = (Q, \Sigma, \delta, q_i, Q_m)$. A marked realisation $\bar{G}_i(p_i)$ is a deterministic labelled transition system $\mathcal{T} = (\bar{Q}, \Sigma \times \{0, 1\}, \delta, \bar{q}_i)$ such that

1. there exists a surjective function $\text{MERGE} : \bar{Q} \rightarrow Q$ with $\text{MERGE}(\bar{q}_i) = q_i$ (3a)
2. $(\forall \bar{q}_i, \bar{q}_k \in \bar{Q}, \forall \sigma \in \bar{\Sigma}, (\bar{q}_j, (\sigma, 0), \bar{q}_k) \in \delta) \Rightarrow (\forall \bar{q}_r \in \bar{Q}, (\bar{q}_j, (\sigma, 1), \bar{q}_r) \notin \delta)$ (3b)
3. $(\forall \bar{q}_i, \bar{q}_k \in \bar{Q}, \forall \sigma \in \bar{\Sigma}, (\bar{q}_j, (\sigma, 1), \bar{q}_k) \in \delta) \Rightarrow (\forall \bar{q}_r \in \bar{Q}, (\bar{q}_j, (\sigma, 0), \bar{q}_r) \notin \delta)$ (3c)
4. $\forall \bar{q}_i, \bar{q}_k \in \bar{Q}, \sigma \in \bar{\Sigma}$, we have

$$(\bar{q}_j, (\sigma, 0), \bar{q}_k) \in \delta$$

$$\Rightarrow \exists h \in L_m(G_i) (p_i(h) = \epsilon) \setminus (\text{MERGE}(\bar{q}_k) = \delta(q_i, h), \sigma))$$

Thus, under state-dependent unobservability, an event $\sigma \in \Sigma$ may be observable at one state $q_i \in Q$ and unobservable at some other state $q_k \in Q$, but the observability of $\sigma$ is not dependent on how those states are reached.
\[(q_i, \sigma, 1), \tilde{q}_k) \in \tilde{\delta} \]
\[\Leftrightarrow \exists h \in L_m(G_i) \left( p_i^h(\sigma) = \sigma \cap (\text{Merge}(\tilde{q}_i) = \delta(q_i, h) \cap (\text{Merge}(\tilde{q}_k) = \delta(\delta(q_i, h), \sigma)5). \right) \]

(3d)

The function \text{Merge} defines how the states of the transition system may be merged to yield the underlying plant states. Hence we refer to it as the merging map. Note that Property 4 in Definition 11 has the following implication: While it is possible that \(p_i^h(\sigma) = \epsilon\) and \(p_i^j(\sigma) = \sigma\) with \(\delta(q_i, s) = \delta(q_i, t) = q_j\) in the underlying plant \(G_n\), the state \(q_j\) must be split into two or more states in a corresponding marked realisation to ensure that the strings \(s\) and \(t\) terminate on different states in \(\langle G_n, p_i \rangle\). We formalise this observation in the following lemma:

**Lemma 1:** For a given unobservability map \(p_i\), a DFSA \(G_i = (Q, \Sigma, \delta, q_i, Q_m)\) and a marked realisation \(\langle G_n, p_i \rangle\) with the merging map \text{Merge}, we have

\[ \exists s, t \in L_m(G_i) \left( \exists \sigma \in \Sigma, p_i^j(\sigma) = \epsilon \cap (\text{Merge}(\tilde{q}_i) = \delta(q_i, s) = q_j \in Q) \right) \]
\[ \Rightarrow \text{Card}(\text{Merge}^{-1}(\{q_j\})) > 1. \] (4)

**Proof:** Denote \((G_i, p_i) = (\hat{Q}, \Sigma \times \{0, 1\}, \hat{\delta}, \hat{q}_i)\) and let \(\exists s, t \in L_m(G_i) (p_i^j(\sigma) = \epsilon \cap (p_i^j(\sigma) = \sigma \cap \delta^*(q_i, s) = \delta^*(q_i, t) = q_j \in Q)\). Since \text{Merge} is onto, we have \(\forall q_m \in Q, \text{Merge}^{-1}(\{q_m\}) \neq \emptyset\), i.e., \(\text{Card}(\text{Merge}^{-1}(\{q_j\})) > 0\). Assume if possible \(\text{Merge}^{-1}(\{q_j\}) = \{\tilde{q}_k\}\). Since \(\delta^*(q_i, s), \sigma\) is defined, surjectivity of \text{Merge} implies \(\exists \tilde{q}_k \in \tilde{Q}\) such that \(\text{Merge}(\tilde{q}_k) = \delta(q_i, s, \sigma)\). It follows that \((p_i^j(\sigma) = \epsilon \cap (\text{Merge}(\tilde{q}_k) = q_j = \delta^*(q_i, s)) \cap (\text{Merge}(\tilde{q}_k) = \delta^*(q_i, s, \sigma))\) is true, implying that \((\tilde{q}_k, (\sigma, 0), \tilde{q}_j) \in \tilde{\delta}\) from Property 4 of Definition 11. Similarly, starting with \(t\), we can argue that \(\exists q_m \in Q\) such that \(\tilde{q}_k, (\sigma, 1), \tilde{q}_j\) \(\in \tilde{\delta}\). This contradicts Property 2 of Definition 11 implying \(\text{Card}(\text{Merge}^{-1}(\{q_j\})) > 1\). \(\square\)

Example 1 illustrates that there exist situations where no finite marked realisation can be derived.

Definition 11 provides a more general approach to partitioning the event alphabet \(\Sigma\) into unobservable events \(\Sigma_u\) and observable events \(\Sigma_o\). The observability of an event from a particular state is made dependent on how the state is reached. A particular event \(\sigma\), defined at state \(q_k\), may be observable for some paths reaching the state \(q_k\) and unobservable for others. Consequently, the state \(q_k\) must be split in a marked realisation and there will be some transitions labelled \((\sigma, 0)\) and some as \((\sigma, 1)\) in \(\langle G_n, p_i \rangle\) all of which correspond to \(\sigma\) (at state \(q_k\)) in the underlying plant.

**Example 1:** It follows from the DFSA in Figure 3 that \(L_m(G_i) \equiv \{0, 1\}^*\). The unobservability map \(p_i\) is specified as follows:

If the number of 1s that occur before the first 0 is \(n\), then \(n\) consecutive 0s thereafter are not observed.

A marked realisation of the DFSA under such an unobservability map is shown in Figure 4. It is shown later in section 4 that such an unobservability specification requires the minimal marked realisation to be infinite and hence no finite marked realisation exists. Note that, for each string in \(1^*0^*\) with \(n \in \mathbb{N}\), the image under the map \(p\) has to be specified separately. Intuitively, this precludes the possibility of existence of a finite marked realisation. Figure 4 shows an example of infinite marked realisation. The observed language \(O_p\) is regular. In fact, \(P : \{0, 1\}^* \rightarrow \{0, 1\}^*\) is surjective and non-injective.

A marked realisation is, in general, not unique. For a given DFSA and a specified unobservability
map, there can be at most countable number of distinct marked realisations. However, it will be shown that the minimal marked realisation (in a sense to be clarified in Proposition 2) is unique. The proof requires the notion of projective Nerode equivalence.

Definition 12: For a given alphabet $\Sigma$, the Nerode equivalence relation $(N)$ on $\Sigma^*$ induced by a language $L$ is defined as $\forall x, y \in \Sigma^* (xN y \iff (\forall u \in \Sigma^* (xu \in L) \iff (yu \in L)))$. The Nerode equivalence relation is an example of a right invariant relation. A language $L$ is regular if and only if the corresponding Nerode equivalence relation is of finite index.

Remark 2: The Nerode equivalence relation induced on $\Sigma^*$ by a given DFSA $G_i = (Q, \Sigma, \delta, q_i, Q_m)$ is the Nerode equivalence on $\Sigma^*$ induced by the marked or accepted language $L_m(G_i)$ (see section 2C).

Definition 13: Projective Nerode equivalence $(N_p)$ with respect to an unobservability map $p$ on a language $L$, is defined as $\forall x, y \in \Sigma^* (xN_p y \iff (\forall u \in \Sigma^* (xu \in L) \Rightarrow (p_i(u) = p_i(u))))$.

Lemma 2: Given a DFSA $G_i = (Q, \Sigma, \delta, q_i, Q_m)$ and an unobservability map $p, N_p$ is a right invariant equivalence relation on $\Sigma^*$, i.e., $\forall x, y, u \in \Sigma^* (xN_p y \Rightarrow xuN_p yu)$. 

Proof: Let $x, x', u \in \Sigma^*$ such that $xN_{p} y$. Since, Nerode equivalence $N_p$ is right invariant, we have $(xN_p y) \Rightarrow (x'N_p y)$ if $xu \in \overline{L_m(G_i)}$. Thus, $xN_p y$. Assume $xu \in \overline{L_m(G_i)}$. Let $r \in \Sigma^*$ such that $xu \in \overline{L_m(G_i)}$. Then $(xN_p y) \Rightarrow (x'N_p y)$ implies $p_i(u) = p_i(u)$ which in turn implies $p_i(u^R) = p_i(u^R)$. Hence we have $(xuN_p yu) \Rightarrow (x'N_p yu)$ if $xu \notin \overline{L_m(G_i)}$, then $\forall v \in \Sigma^* (xuvN_p yuv)$. This completes the proof. \qed 

To show that minimal marked realisations are unique we need the following definitions.

Definition 14: Let $(G_i, p_i) = (Q, \Sigma \times \{0, 1\}, \tilde{\delta}, \tilde{q}_i)$. Further, let $q_d$ be a symbol not in $\tilde{Q}$. Then $\tilde{\delta}^* : \tilde{Q} \cup \{q_d\} \times (\Sigma \times \{0, 1\})^* \rightarrow \tilde{Q} \cup \{q_d\}$ is defined recursively as

$$\forall q \in \tilde{Q} \bigcup \{q_d\}, \tilde{\delta}^*(q, \epsilon) = q$$

$$(5a)$$

$$\forall q \in \tilde{Q} \bigcup \{q_d\}.$$

Lemma 3: Following Definition 11, let $(G_i, p_i) = (\tilde{Q}, \Sigma \times \{0, 1\}, \tilde{\delta}, \tilde{q}_i)$ be a marked realisation. Then $\forall x \in \Sigma^* \tilde{\delta}^*(\tilde{q}_i, \phi(x)) = \tilde{q}_i \neq q_d \Rightarrow x \in L(q_i, \text{Merge}(\tilde{q}_i)), (8)$$

where $q_d$ and $\phi$ are given in Definitions 14 and 15.

Proof: For $|x| = 0$, the result follows from Definitions 14 and 15 since $\tilde{\delta}^*(\tilde{q}_i, \phi(\epsilon)) = \tilde{\delta}^*(\tilde{q}_i, \epsilon) = \tilde{q}_i$, and $\epsilon \in L(q_i, \phi(\epsilon))$. Hence we consider $|x| \geq 1$. We use induction on the length of $x$. For $|x| = 1$, we note that $\tilde{\delta}^*(\tilde{q}_i, \phi(x)) = \tilde{q}_i \neq q_d$ implies that either $(\tilde{q}_i, (x, 0), \tilde{q}_i) \in \delta$ or $(\tilde{q}_i, (x, 1), \tilde{q}_i) \in \delta$. In either case, Definition 11 implies $\delta(q_i, x) = \text{Merge}(\tilde{q}_i)$ in turn implies $x \in L(q_i, \text{Merge}(\tilde{q}_i))$. 

Next we show that if Equation (8) is true for $|x| = r \in \mathbb{N}$, then it is also true for $x\sigma$ where $\sigma \in \Sigma$. Let $|x| = r \in \mathbb{N}$ and $\tilde{\delta}^*(\tilde{q}_i, \phi(x)) = \tilde{q}_i \neq q_d$. Then we have $\tilde{\delta}^*(\tilde{q}_i, \phi(x), \sigma) = \tilde{q}_i \neq q_d$. Since $\tilde{\delta}^*(\tilde{q}_i, \tau) = q_d$ for all $\tau \in (\Sigma \times \{0, 1\})^*$ and $q_d \neq q_d$, it follows that

$$\tilde{\delta}^*(\tilde{q}_i, \phi(x), \sigma) = \tilde{q}_i \neq q_d$$

$$(9)$$

$$\Rightarrow \tilde{\delta}^*(\tilde{q}_i, \phi(x), \sigma) = \tilde{q}_i \neq q_d$$

$$(10)$$

Equation (11) follows from the same argument as given for the case $|x| = 1$. Since $|x| = r$, Equation (9) implies

$$x \in L(q_i, \text{Merge}(\tilde{q}_i)).$$

$$(12)$$

Finally, it follows from Equations (11) and (12) that $x \sigma \in L(q_i, \text{Merge}(\tilde{q}_i))$. This completes the proof. \qed

Lemma 4: Following Definition 11, let $(G_i, p_i) = (\tilde{Q}, \Sigma \times \{0, 1\}, \tilde{\delta}, \tilde{q}_i)$ be a marked realisation. Then

$$\forall x \in \Sigma^* (x \in \overline{L_m(G_i)} \Rightarrow \tilde{\delta}^*(\tilde{q}_i, \phi(x)) \neq q_d), (13)$$

Note that Property 2 of Definition 11 (determinism) is necessary for Equation 5b to make sense, i.e., to guarantee that there is at most one $\tilde{q}_i$ that satisfies $(\tilde{q}_i, \tau, \tilde{q}_i) \in \delta$.

Definition 15: There exists an one-to-one map $\phi : \Sigma^* \rightarrow (\Sigma \times \{0, 1\})^*$ given by

$$\phi(\epsilon) = \epsilon$$

$$(6)$$

$$\phi(\phi(x)) = \begin{cases} \phi(\phi(\sigma), 0), & \text{if } p_i(\sigma) \text{ is defined and } p_i(\sigma) = \epsilon \\ \phi(\phi(\sigma), 1), & \text{otherwise} \end{cases}$$

$$(7)$$

\[\text{Lemma 3: Following Definition 11, let } (G_i, p_i) = (\tilde{Q}, \Sigma \times \{0, 1\}, \tilde{\delta}, \tilde{q}_i) \text{ be a marked realisation. Then } \forall x \in \Sigma^* (\tilde{\delta}^*(\tilde{q}_i, \phi(x)) = \tilde{q}_i \neq q_d \Rightarrow x \in L(q_i, \text{Merge}(\tilde{q}_i)), (8)\]
where \( q_d \) and \( \phi \) are given in Definitions 14 and 15.

**Proof:** For \( |x| = 0 \), the result follows from noting 
\[
\delta^*(\tilde{q}_i, \phi(e)) = \delta^*(\tilde{q}_i, \epsilon) = \tilde{q}_i \neq q_d \text{ and } e \in L_m(G_i).\]
Hence, we consider \( |x| \geq 1 \).

**(Left to Right)** Assume \( x \in L_m(G_i) \). We use induction on the length of \( x \). Let \( |x| = 1 \). We have \( \delta(q_i, x) = q_k \) for some \( q_k \in Q \). Since \( \text{MERGE} \) is onto, Definition 11 implies either \((\tilde{q}_i, (x, 0), q_1^t) \in \delta \) or \((\tilde{q}_i, (x, 1), q_2^t) \in \delta \) for some \( q_1^t \in \tilde{Q} \) with \( \text{MERGE}(q_1^t) = q_k \). In either case, we have \( \delta^*(\tilde{q}_i, \phi(x)) = q_k \neq q_d \).

Next we show that if Equation (13) is true from left to right for \( |x| = r \in \mathbb{N} \), then it is also true for \( x \sigma \) where \( \sigma \in \Sigma \). To complete the induction, we choose an arbitrary \( x \in L_m(G_i) \) with \( |x| = r \in \mathbb{N} \) such that \( \delta^*(\tilde{q}_i, \phi(x)) = \tilde{q}_k \neq q_d \). Then for \( \sigma \in \Sigma \) such that \( x \sigma \in L_m(G_i) \), we have
\[
\delta^*(\tilde{q}_i, \phi(x) \sigma) = \delta^*(\tilde{q}_i, \phi(x)) \neq q_d.
\]

The last step is based on the same reasoning as for \( |x| = 1 \), except \( q_i \) is replaced by \( q_k \).

**(Right to Left)** As before, we use induction on the length of \( x \). Let \( |x| = 1 \) and \( \delta^*(\tilde{q}_i, \phi(x)) = \tilde{q}_k \neq q_d \). It follows from Definition 14 that \( \tilde{q}_i, \phi(x), \tilde{q}_k \in \delta \) which in turn implies that either \((\tilde{q}_i, (x, 0), q_1^t) \in \delta \) or \((\tilde{q}_i, (x, 1), q_2^t) \in \delta \) and hence \( \delta(\text{MERGE}(\tilde{q}_i), x) = \text{MERGE}(\tilde{q}_i) \).

Also, based on the reasoning for the case \( |x| = 1 \), it follows from Equations (14) and (15) that
\[
\delta^*(\tilde{q}_i, \phi(\sigma)) = \tilde{q}_k \neq q_d \Rightarrow \sigma \in L_m(G_i) \tag{17}
\]

Finally, it follows from Equations (16) and (17) that \( x \sigma \in L_m(G_i) \). This completes the proof.

**Proposition 1:** Every marked realisation \((G_i, p_i) \) = \((\tilde{Q}, \Sigma \times \{0, 1\}, \delta, \tilde{q}_i) \) for a given DFSA \( G_i \) with a specified unobservability map \( p_i \) induces a right invariant equivalence relation \( \mathcal{R}_{(G_i, p_i)} \) on \( \Sigma^* \) defined by
\[
\forall x, y \in \Sigma^* \text{ s.t. } x \mathcal{R}_{(G_i, p_i)} y \Leftrightarrow (\delta^*(\tilde{q}_i, \phi(x)) = \delta^*(\tilde{q}_i, \phi(y))).
\]

**Proof:** It is readily seen that \( \mathcal{R}_{(G_i, p_i)} \) is an equivalence relation on \( \Sigma^* \). For right invariance (Hopcroft et al. 2001), we need to show that
\[
\forall \omega \in \Sigma^* \text{ s.t. } \delta^*(\tilde{q}_i, \phi(x\omega)) = \delta^*(\tilde{q}_i, \phi(y\omega)). \tag{19}
\]

For \( |\omega| = 0 \), the result is immediate. Hence we consider \( |\omega| \geq 1 \). First we note that \( \delta^*(\tilde{q}_i, \phi(x)) = \delta^*(\tilde{q}_i, \phi(y)) \) implies (see Definition 14) that either \( \delta^*(\tilde{q}_i, \phi(x)) = \delta^*(\tilde{q}_i, \phi(y)) = \tilde{q}_k \) for some \( \tilde{q}_k \in \tilde{Q} \) or \( \delta^*(\tilde{q}_i, \phi(x)) = \delta^*(\tilde{q}_i, \phi(y)) = q_d \). If \( \delta^*(\tilde{q}_i, \phi(x)) = \delta^*(\tilde{q}_i, \phi(y)) = \tilde{q}_k \), then Equation (5c) in Definition 14 implies that \( \forall \omega \in \Sigma^* \text{ s.t. } \delta^*(\tilde{q}_i, \phi(x)) = \tilde{q}_k \). Hence we assume \( \delta^*(\tilde{q}_i, \phi(x)) = \delta^*(\tilde{q}_i, \phi(y)) = \tilde{q}_k \). We use induction on the length of \( \omega \). Let \( |\omega| = 1 \). Definition 11 implies \( \delta(q_i, x) = \text{MERGE}(\tilde{q}_i) = \delta(q_i, y) \) and hence \( (x\omega, y\omega) \in L_m(G_i) \) or \( x, y \in L_m(G_i) \).

If \( x\omega, y\omega \not\in L_m(G_i) \), then from Lemma 4, we have
\[
\delta^*(\tilde{q}_i, \phi(x\omega)) = \delta^*(\tilde{q}_i, \phi(y\omega)) = \delta^*(\tilde{q}_i, \phi(x\omega)) = \delta^*(\tilde{q}_i, \phi(y\omega)) \tag{20}
\]

Otherwise if \( x\omega, y\omega \in L_m(G_i) \) we have
\[
\delta(q_i, x\omega) = \delta(q_i, y\omega) = \delta(\text{MERGE}(\tilde{q}_i), \omega) = q_d \in \tilde{Q}. \tag{21}
\]

Now, we claim \( \phi(x\omega) = \phi(x) \tau \) and \( \phi(y\omega) = \phi(\gamma) \tau \) where \( \tau = |(\omega, 0), (\omega, 1) \rangle \). The argument is as follows: Assume if possible \( \phi(x\omega) = \phi(x)(\omega, 0) \) and \( \phi(y\omega) = \phi(\omega)(\omega, 1) \).

Since both \( p_i^1(\omega) \) and \( p_i^2(\omega) \) are defined (due to \( x\omega, y\omega \in L_m(G_i) \)), we have from Definition 15 that \( p_i^1(\omega) = \epsilon \) while \( p_i^2(\omega) = \omega \). It follows that both \( (\tilde{q}_i, (\omega, 0), q_1) \in \delta \) and \( (\tilde{q}_i, (\omega, 1), q_2) \in \delta \) for some \( q_1, q_2 \in \text{MERGE}^{-1}(|(\omega, 0), (\omega, 1) \rangle) \) which contradicts Properties 2 and 3 of Definition 11. Now \( q_1, q_2 \) are not necessarily distinct. Therefore, we have \( \phi(x\omega) = \phi(x) \tau \) and \( \phi(y\omega) = \phi(\gamma) \tau \) implying that
\[
\delta^*(\tilde{q}_i, \phi(x\omega)) = \delta^*(\tilde{q}_i, \phi(y\omega)). \tag{22}
\]

Equations (20) and (22) imply that Equation (19) is true for \( |\omega| = 1 \). Next we show that if Equation (19) is true for all \( \omega \in \Sigma^* \) with \( |\omega| = r \in \mathbb{N} \), then it is also true for \( r \sigma \) where \( \sigma \in \Sigma \). Let \( |\omega| = r \in \mathbb{N} \). Then we have \( \delta^*(\tilde{q}_i, \phi(x\omega\sigma)) = \delta^*(\tilde{q}_i, \phi(x\omega\sigma)) = \delta^*(\tilde{q}_i, \phi(\gamma\omega\sigma)) \). This completes the proof.

**Proposition 2:** For a marked realisation \((G_i, p_i) \) of a DFSA \( G_i \) with a specified unobservability map \( p_i \),

1. \( \mathcal{N}_p \) is a refinement of \( \mathcal{N} \), i.e., \( \mathcal{N}_p \leq \mathcal{N} \).
2. If \( (G_i, p_i) \) is a marked realisation of the plant automaton \( G_i \) with an unobservability map \( p_i \), then \( \mathcal{R}_{(G_i, p_i)} \leq \mathcal{N}_p \).
3. \( \mathcal{N}_p \) induces a marked realisation \((G_i, p_i)^* \) with \( \mathcal{R}_{(G_i, p_i)^*} = \mathcal{N}_p \).
4. \( (G_p,p)_* \) is the unique (up to a renaming of states) minimal marked realisation in the sense \( R_{(G_p,p)} \leq R_{(G_p,p)_*} \).

**Proof:**

1. It follows from Definitions 12 and 13 that \( N'_p \) is a refinement of \( N \).

2. We need to show

\[ \forall x,y \in \Sigma^* (xR_{(G_p,p)_*}y \Rightarrow xN'_p y) \]

i.e. \( \forall x,y \in \Sigma^* (\delta^*(q_i,\phi(x)) = \tilde{\delta}^*(q_i,\phi(y))) \Rightarrow xN'_p y \)

Let \( x,y \in \Sigma^* \) with \( \tilde{\delta}^*(q_i,\phi(x)) = \tilde{\delta}^*(q_i,\phi(y)) \). As before, two cases are possible: either (1) \( \tilde{\delta}^*(q_i,\phi(x)) = \tilde{\delta}^*(q_i,\phi(y)) = \tilde{q}_k \) for some \( q_k \in \tilde{Q} \) or (2) \( \tilde{\delta}^*(q_i,\phi(x)) \neq \tilde{\delta}^*(q_i,\phi(y)) = q_k \). Note that \( \text{MERGE}(q_i) = q_i \). For for Case (1), Lemma 3 implies

\[ \delta(q_i,x) = \delta(q_i,y) = \text{MERGE}(q_k) \in \tilde{Q} \Rightarrow xN'_p y. \]  

(23)

Now, we claim

\[ \forall u \in \Sigma^* (xu \in L_m(G_i) \Rightarrow p^*_i(u) = p^*_j(u)). \]  

(24)

For \( |u| = 0 \), the result is immediate from noting \( p^*_i(e) = p^*_j(e) = e \). Hence we consider \( |u| \geq 1 \). We use induction on the length of \( u \). Let \( |u| = 1 \). Assume \( xu \in L_m(G_i) \) and, if possible, let \( p^*_i(u) = xu \Rightarrow p^*_j(u) = y \).

Since \( \delta^*(q_i,x) = \delta^*(q_i,y) = \text{MERGE}(q_k) \) and \( xu \in L_m(G_i) \), there exists \( x \in \tilde{Q} \) such that \( \delta(\text{MERGE}(q_k),u) = q_k \).

Hence, it follows that \( (q_k(u),q_k(u_1),q_k(u_2)) \in \tilde{E} \Rightarrow (q_k(u),q_k(u_1),q_k(u_2)) \in \delta \) where \( \text{MERGE}(q_k(u_1)) = \text{MERGE}(q_k(u_2)) = q_k \). This contradicts Properties 2 and 3 in Definition 11. Next we show that if Equation (24) is true for all \( u \in \Sigma^* \) such that \( |u| = r \in \mathbb{N} \), then it is also true for \( u \sigma \) where \( \sigma \in \Sigma \). Let \( |u| = r \in \mathbb{N} \) and \( xu = \text{MERGE}(q_k) \). Then we have

\[ p^*_i(u \sigma) = p^*_i(u)p^*_i(u \sigma) = p^*_j(u)p^*_j(u \sigma). \]  

(25)

Since \( R_{(G_p,p)_*} \) is right invariant (see Proposition 1), we have \( \delta^*(q_i,\phi(xu)) = \delta^*(q_i,\phi(yu)) \). Hence it follows from the same argument as given for the case \( |u| = 1 \), that \( p^*_i(u \sigma) = p^*_j(u \sigma) \). Hence, we have

\[ p^*_i(u \sigma) = p^*_i(u)p^*_i(u \sigma) = p^*_i(u)p^*_j(u \sigma) = p^*_i(u \sigma). \]  

(26)

This completes the induction. Equations (23) and (26) implies \( xN'_p y \).

For Case (2), Lemma 4 implies \( x,y \notin L_m(G_i) \). Prefix closure of \( L_m(G_i) \) implies \( \forall u \in \Sigma^* (xu,u \notin L_m(G_i)) \) which in turn implies \( xN'_p y \).

Since the condition \( \forall u \in \Sigma^* (xu \in L_m(G_i) \Rightarrow p^*_i(u) = p^*_j(u)) \) is vacuously satisfied (since there is no such \( u \)), we have \( xN'_p y \). This completes the proof.

3. Let \( \delta^p = \{ E_k : k \in J \subseteq \mathbb{N} \} \) be the set of equivalence classes of \( N'_p \) where \( J \) is an appropriate index set. We construct a labelled deterministic transition system \( (G_i,p)_* = (\delta^p, \Sigma \times [0,1], \tilde{\delta}, \tilde{q}_i) \) as follows:

- \( \tilde{q}_i = E_i \), where \( \epsilon \in E_i \in \delta^p \).
- \( \exists x \in \Sigma^* ((x \in E_k) \land (x \sigma \in E_j) \land (p^*_i(\sigma) = \epsilon)) \Rightarrow (E_k, (\sigma,0), E_j) \in \tilde{\delta} \).
- \( \exists x \in \Sigma^* ((x \in E_k) \land (x \sigma \in E_j) \land (p^*_j(\sigma) = \epsilon)) \Rightarrow (E_k, (\sigma,1), E_j) \in \tilde{\delta} \).
- There are no other elements in \( \tilde{\delta} \).

Next we show that \( (G_i,p)_* \) is a marked realisation in the sense of Definition 11. Since \( N'_p \leq N \), there exists a surjective map \( \Psi: \delta^p \rightarrow Q \). This serves as the merging map for \( (G_i,p)_* \). Next we claim \( \exists x \in E_k ((x \in E_j) \land (p^*_i(\sigma) = \epsilon)) \Rightarrow \forall x \in E_k ((x \in E_j) \land (p^*_j(\sigma) = \epsilon)) \). The argument is as follows: Let \( x, y \in E_k \) with \( x \neq y \) and \( x \sigma \in E_j \). It follows form right invariance of \( N'_p \) (See Lemma 2) that \( x \sigma N'_p y \Rightarrow y \sigma E_j \). Also, \( xN'_p y \Rightarrow p^*_i(\sigma) = p^*_j(\sigma) \). Hence we conclude \( (E_k, (\sigma,1), E_j) \in \tilde{\delta} \Rightarrow (E_k, (\sigma,1), E_j) \notin \delta^p \). Similarly, \( (E_i, (\sigma,1), E_j) \in \tilde{\delta} \Rightarrow (E_i, (\sigma,0), E_j) \notin \delta^p \). Thus Properties 2 and 3 of Definition 11 are satisfied. Property 4 is satisfied by construction as follows. We note that for \( h \in L_m(G_i) \), \( \sigma \in \Sigma \), \( \delta(\tilde{q}_i,h) = \delta(q_i,h) \) \land \( \Psi(E_i) = \delta(\tilde{q}_i,h) \) \land \( \Psi(E_j) = \delta(q_i,h) \) \land \( p^*_i(\sigma) = p^*_j(\sigma) \). Hence we have \( (E_k, (\sigma,0), E_j) \in \tilde{\delta} \). Similarly, if \( p^*_i(\sigma) = \sigma \) then \( (E_k, (\sigma,1), E_i) \in \tilde{\delta} \). Thus \( (G_i,p)_* \) is indeed a marked realisation for plant \( G_i \) with the unobservability \( p_i \).

Finally, we show \( R_{(G_i,p)_*} = N'_p \). From since item (2) above we have \( R_{(G_i,p)_*} \leq N'_p \), we only need to show the converse, i.e., to show \( N'_p \leq R_{(G_i,p)_*} \) for which we need to prove the following statement:

\[ xN'_p y \Rightarrow (\delta^*(\tilde{q}_i, \phi(x)) = \delta^*(\tilde{q}_i, \phi(y))). \]  

(27)

First, we claim that

\[ x \in E_k \Rightarrow \delta^*(\tilde{q}_i, \phi(x)) = E_k. \]  

(28)

We proceed by the method of induction. The result is immediate for \( |x| = 0 \), i.e., \( x = \epsilon \) from Definitions 14 and 15 by noting

\[ \delta^*(\tilde{q}_i, \phi(\epsilon)) = \delta^*(\tilde{q}_i, \epsilon) = \tilde{q}_i = E_i. \]  

(29)

\( \epsilon \in E_i \) (By Definition of \( \delta^p \) above).

(30)

We assume that Equation (28) holds for all \( x \in \Sigma^* \) with \( |x| \leq n \) for some \( n \in \mathbb{N} \). The induction is then completed by noting that for any string \( x \sigma \) with \( |x| = n \) and \( \sigma \in \Sigma \), we have

\[ x \in E_k \Rightarrow \delta^*(\tilde{q}_i, \phi(x)) = E_k \]  

(31)

\[ (\sigma \epsilon \in E_j \Rightarrow \forall y \in E_k (y \sigma \in E_j)) \]  

(32)

\[ \Rightarrow \delta^*(\tilde{q}_i, \phi(x \sigma)) = \delta^*(\delta^*(\tilde{q}_i, \phi(x)), \sigma)) = \delta^*(E_k, \phi(\sigma)) = E_j. \]
Now, let \(x, y \in \Sigma^*\) such that \(xN_p'y\), i.e., 
\(\forall N y \sqcup (\forall u \in \Sigma^*(xu \in T \Rightarrow (p_i'(u) = p_i'(u)))\).
It follows that \(x, y \in E_k \in \mathcal{E}_{p'}\) which in turn implies from Equation (28) that \(\delta^*(q_i, \phi(y)) = \delta^*(q_i, \phi(x))\) and hence we have \(xR_{(G_i, p_i)}y\). This completes the proof.

4. It is immediate from items (2) and (3) above that for any arbitrary marked realisation \(\langle G_i, p_i \rangle\), we have \(R_{(G_i, p_i)} \leq R_{(G_i, p_i)}^*:\). Assume there exists a marked realisation \(\langle G_i, p_i \rangle\) such that \(R_{(G_i, p_i)} = R_{(G_i, p_i)}^*:\). From item (3) it follows \(R_{(G_i, p_i)} = N_p'\). Hence there exists a bijective mapping between the equivalence classes of \(R_{(G_i, p_i)}\) and that of \(N_p'\). Hence, \(\langle G_i, p_i \rangle^*\) is the minimal realisation unique up to renaming of states.

\[\square\]

**Definition 16:** The minimal marked realisation \(\langle G_i, p_i \rangle^*\) of a given DFSA \(G_i\) for an unobservability map \(p_i\) is referred to as the \(p\)-minimal realisation in the sequel.

For state-independent unobservability in Figure 1, the alphabet \(\Sigma\) can be partitioned into observable and unobservable parts. Consequently, the projective Nerode equivalence becomes equivalent to Nerode equivalence. This is also true for state-dependent unobservability because the observability of a symbol is only dependent on the state of the automaton, from which it is generated. This fact is restated in the following proposition.

**Proposition 3:** For state-dependent unobservability, \(N_p' = N'\).

**Proof:** Let \(E_k\) be an equivalence class of the Nerode equivalence relation \(N\). State-dependent unobservability implies that if \(\sigma \in \Sigma\) is defined from the state corresponding to the equivalence class \(E_k\), \(\forall x, y \in E_k\), \(p_i'(\sigma) = p_i'(\sigma)\). Let the event \(\sigma\) lead the automaton to a state that corresponds to the equivalence class \(E_j\). Then, \(\forall x, y \in E_k\), \(p_i'(\sigma) = p_i'(\sigma)\) \(\forall u \in \Sigma^*\) such that \(xu \in L\) and hence, \(xN y \Rightarrow xN_p'y \forall x, y \in L\), \(\Rightarrow N \leq N_p'\).

By Proposition 2, we have \(N_p' \leq N\) \(\Rightarrow N \equiv N_p'\).

**Remark 3:** State-independent unobservability is a special case of state-dependent unobservability and so the result applies to the case of state independent unobservability as well.

If the \(p\)-minimal realisation is infinite, then the projective Nerode equivalence relation is not of finite index. The rationale is that the projective Nerode equivalence \(N_p'\) is a refinement of the Nerode equivalence \(N\). An equivalence class under \(N\) is possibly split further under \(N_p'\). If the \(j\)th equivalence class of \(N\) is denoted by \(N^j\), and the total number of splits that \(N^j\) undergoes under \(N_p'\) is denoted by \([N^j : p]\), then we have:

**Lemma 5:** \(\langle G_i, p_i \rangle^* \uparrow \Leftrightarrow (\exists j \in I, g, p \equiv [N^j : p])\)

**Proof:** \(\langle G_i, p_i \rangle^* \uparrow \Leftrightarrow \exists j \in I(\mathcal{Q}(\{q_j\}) \uparrow) \Leftarrow \Rightarrow \exists j \in I([N^j : p] \uparrow)\)

**Proposition 4:** Let \(\mathcal{Q}\) denote the set of regular languages for an DFSA \(G_i\) and let \(p_i\) be an unobservability map and \(\Omega_p\) be the corresponding observed language. If there exists a finite minimal realisation \(\langle G_i, p_i \rangle\) induced by \(N_p\), then \(\Omega_p\) is regular. That is \(\langle G_i, p_i \rangle \downarrow \Rightarrow \Omega_p \in \mathcal{Q}\).

**Proof:** A finite \(p\)-minimal realisation implies that the observed language is regular. This is a well-known result and the proof is sketched for the sake of completeness. Since \(\langle G_i, p_i \rangle\) is finite, the marked transitions can be replaced by \(\epsilon\)-transitions. The resultant automaton is a finite state machine with \(\epsilon\)-transitions which can be reduced to a DFSA by standard algorithms (Hopcroft et al. 2001).

It follows from Proposition 3 that, under both state-dependent and state-independent unobservability, the observed language is regular. Nevertheless, the observed language may be significantly more complex, with the minimum number of states required for an automaton realisation being a possibly exponential function of the original number of states (Wong 1998).

We show that in more complex cases, the observed language \(\Omega_p\) may be non-regular.

**Definition 17:** For a given DFSA \(G_i\) and an unobservability map \(p_i\), the \(p\)-minimal realisation \(\langle G_i, p_i \rangle^*\) is said to satisfy the condition \(\mathcal{P}_\infty\) if the following statement is true.

\[\forall h \in L_m(G_i), \forall \sigma \in \Sigma, \forall u \in \Sigma^* \quad (\exists \alpha \in L_m(G_i), (p_i'(\sigma) = \epsilon) \Rightarrow (p_i'(\alpha u) = \epsilon))\]

Intuitively, \(\mathcal{P}_\infty\) implies that once an unobservable transition is encountered, all succeeding transitions are unobservable.

**Proposition 5:** \(\langle G_i, p_i \rangle^* \uparrow \land \mathcal{P}_\infty \Rightarrow (\Omega_p \not\in \mathcal{Q})\)

**Proof:** Since the \(p\)-minimal realisation \(\langle G_i, p_i \rangle^*\) is an infinite realisation, it follows from Lemma 5, that \(\exists q_k \in \mathcal{Q}[\{q_k\}]\), i.e., there exists a \(q_k \in G_i\), which is split into (countably) infinite number of states \(\{q_{k1}, q_{k2}, \ldots, q_{kj}, \ldots\}\) in \(\langle G_i, p_i \rangle^*\) as illustrated in Figure 5.

The strings \(h_1, h_2, \ldots, h_j, \ldots\) that are assumed to start from the initial state of the automaton. These strings are chosen so that if they are truly followed in the automaton graph (i.e., without
considering unobservability), then they all lead to the same state \( q_k \) in \( G_i \). This has been illustrated in the lower part of the diagram in Figure 5. Such an infinite set of strings exists; otherwise, the set \( \{q_{k1}, q_{k2}, \ldots, q_{kn}, \ldots\} \) must collapse to a finite set and the condition \( (G_i, p_i)^* \uparrow \) will be violated. The fact that one may choose \( h_1, h_2, \ldots, h_n, \ldots \) follows from the axiom of choice. Now, \( \mathbb{P}_\infty \) implies that at most one string from the set \( \{h_1, h_2, \ldots, h_n, \ldots\} \) may have an unobservable transition. The rationale is presented below.

Let \( h_j \) have at least one unobservable transition. Then, all string suffixes initiating from \( q_k \) is completely unobservable. Now, if there exists another \( h_l \) that has any unobservable transition, with \( h_j \neq h_l \), then the corresponding states \( q_{k1}, q_{k2}, \ldots, q_{kn}, \ldots \) must collapse; this violates the condition that \( (G_i, p_i)^* \) is a minimal realisation. If such a string \( h_j \) exists, which has at least one unobservable transition, this string is deleted from \( \{h_1, h_2, \ldots\} \) and the corresponding state \( q_k \) from \( \{q_{k1}, q_{k2}, \ldots\} \). Let us denote the new string set \( \{h_1, h_2, \ldots\} \) by \( \mathcal{H} \) and the new state set \( \{q_{k1}, q_{k2}, \ldots\} \) by \( \mathcal{Q} \). Note that all elements of \( \mathcal{H} \) are completely observable, i.e., \( p(h) = h_\forall h \in \mathcal{H} \). Next, let us choose two strings \( h_1 \) and \( h_2 \) from \( \mathcal{H} \). Since \( h_1 \) and \( h_2 \) lead to the same state \( q_k \) in the automaton \( G_i \), we have \( h_1 N h_2 \). As \( h_1 \) and \( h_2 \) lead to different states in the \( p \)-minimal realisation \( (G_i, p_i)^* \), it follows \( h_1 N h_2 \).

Now if \( L \) is the accepted language of the automaton, \( (h_1 N h_2) \Rightarrow (\exists u \in \Sigma^* \text{ such that } h_1 u, h_2 u \in L) \) and \( (p_i^h(u) \neq p_i^u(u)) \).

Now, \( \mathbb{P}_\infty \Rightarrow ((p_i^h(u) = \omega_1) \land (p_i^u(u) = \omega_2)) \), where \( u \equiv \omega_1 v_1 \equiv \omega_2 v_2, v_1, v_2 \in \Sigma^* \text{ with } \omega_1 \neq \omega_2 \). This implies that there exists \( \omega_3 \in \Sigma^* \text{ such that either } \omega_2 \equiv \omega_1 \omega_3 \text{ or } \omega_1 \equiv \omega_2 \omega_3 \). Assuming that \( \omega_3 \equiv \omega_1 \omega_3 \), it follows that \( (h_1 \omega_1 \omega_3 \notin \mathcal{Q}) \land (h_2 \omega_1 \omega_3 \notin \mathcal{Q}) \).

By the Nerode equivalence relation, it is concluded that \( h_1 \) and \( h_2 \) lead to distinct states in any realisation for \( \mathcal{Q} \). Since \( h_1 \) and \( h_2 \) are two arbitrary strings in the set \( \mathcal{H} \), each element of \( \mathcal{H} \) must lead to a distinct state in any realisation for \( \mathcal{Q} \). Infinite cardinality of \( \mathcal{H} \) implies \( \mathcal{Q} \notin \text{REG} \).

**Example 2:** This example explains the logic of Proposition 5. The following unobservability is assumed in the automaton of Figure 6.

If \( n \) consecutive 1s are observed, the only suffixes observable thereafter are at most \( n \) 0s.

The \( p \)-minimal realisation is shown in Figure 7, where the unobservability map satisfies the \( \mathbb{P}_\infty \) condition. It is obvious that there is no way to collapse any state in Figure 7. Hence, the \( p \)-minimal realisation is infinite. The observed language \( \mathcal{Q} \) is \( \{1^n 0^m : m \leq n \} \). Hence, \( \mathcal{Q} \notin \text{REG} \) is in accordance with Proposition 5.

Next it is shown that an infinite \( p \)-minimal realisation does not guarantee that the observed language \( \mathcal{Q} \) is non-regular.

**Definition 18:** For a given DFSA \( G_i \) and an unobservability map \( p_i \), the \( p \)-minimal realisation \( (G_i, p_i)^* \) is said to satisfy the condition \( \mathbb{P}_\infty \) if \( p_i \) satisfies the following criterion:

\[
\forall h \in \overline{m}(G), \forall \sigma \in \Sigma, (p_i^h(\sigma) = \varepsilon) \Rightarrow (\exists \forall v_1 \in \Sigma^*, \tau \in \Sigma, \text{ s.t.} (p_i^h (v_1) = \varepsilon) \land (p_i^h (\tau) = \tau) \land (h N h_\forall v_1)).
\]
Remark 4: The condition $P_{\text{Loop}}$ implies that, under the unobservability map $P$, only loops can be unobservable. (Note: A loop means a string $l$ such that $\forall \omega \in \Sigma^*$, starting from the initial state $q_1$, satisfies the condition $\omega \mathcal{N} \omega \mathcal{L}$.) This condition follows from the requirement of $P_{\text{Loop}}$ that the first observable event after an unobservable string $s$ begins from the same state that $s$ is initiated from. Furthermore, the unobservable string $s$ cannot be unbounded. This implies that as a plant traverses a loop $l$, only a bounded number of such successive traversals can be unobservable. If an arbitrary number of successive traversals is unobservable, then $P_{\text{Loop}}$ is violated (i.e., there may not exist a $\tau$ that can be checked from Definition 18). This concept is illustrated in Figure 8.

**Proposition 6:** Let $G_i$ be a DFSA with a specified unobservability map $P_i$. Then, $P_{\text{Loop}} \iff (O_P \equiv L_{\text{min}})$.

**Proof:** To prove $P_{\text{Loop}} \Rightarrow (O_P \equiv L_{\text{min}})$, we proceed as follows. Let $L_{\text{min}}(G_i) = L$. It is given that $P_{\text{Loop}} \Rightarrow (\forall u \in \Sigma^*, p(u) \equiv P(o_o \mathcal{F} \omega = o_o \mathcal{F} \omega \in O_P)$, with $m \geq n \in \mathbb{N}$), where $o_o \mathcal{F} \omega, l \in \Sigma^*$, and $l$ is a loop. Since $l$ is a loop, $(o_o \mathcal{F} \omega \in L) \Rightarrow (o_o \mathcal{F} \omega \in L \forall k \leq m) \Rightarrow (o_o \mathcal{F} \omega \in L)$. Hence, $(p(u) \in O_P) \Rightarrow (p(u) \in L) \Rightarrow (O_P \equiv L)$. Also, $((o_o \mathcal{F} \omega \in L) \cap (o_o \mathcal{F} \omega \in L)) \cap P_{\text{Loop}} \Rightarrow (\exists k \in \mathbb{N} o_o \mathcal{F} \omega \in L)$. Hence, by induction, $k_i \in \mathbb{N}$ such that $(o_o \mathcal{F} \omega \in L) \Rightarrow (u \in O_P)$, i.e., $L \equiv O_P$. Therefore $L \equiv O_P$.

To prove the converse, let $O_P \equiv L$, and let us assume $P_{\text{Loop}}$. The scenario #1 is depicted in Figure 9 where $u_1$ is unobservable and $v_1$ is observable. Furthermore, it is assumed, without loss of generality, that $h_1$ is observable, where $h_1$ starts from the initial state of the plant. It follows that $p(h_1, l_{h_1} v_1) = h_1 v_1 \in O_P$. Now, $O_P \equiv L$ implies that $h_1 v_1 \in L$. Hence, there must be a loop $v_1$ at the state $q_{k_1}$, as seen in the scenario #2 in Figure 10. The same argument is applicable for any other string $\omega_2$ initiating from state $q_{k_2}$. Hence, it follows that $\forall \omega \in \Sigma^* (h_1, u_1 \omega \in L) \iff (h_1 \omega \in L)$. Hence, $q_{k_1}$ and $q_{k_2}$ collapse to the same state in the minimal realisation of the language $L$, which implies that $u_1$ is a loop. The entire loop $u_1$ cannot be unobservable because then $O_P \not\equiv L$. Hence, $(O_P \equiv L) \implies P_{\text{Loop}}$.

**Example 3:** This example illustrates the concept of Proposition 6. For the finite state automaton in Figure 6, let us consider the following unobservability scenario:

If $n$ consecutive 1s are observed, $n$ successive 0s (if generated) are unobservable.

Note that the unobservability map is different from the one in Example 2 although the automaton graph is identical. The unobservability map in this case satisfies $P_{\text{Loop}}$. Since there is no way to collapse any state in the marked realisation shown in Figure 11, the $p$-minimal realisation is infinite. It is easy to verify that the observed language $O_P \equiv \Sigma^*$. Hence $O_P \equiv L$, the plant language, in accordance with Proposition 6.

**Remark 5:** Since the proof of Proposition 6 is independent of the finiteness of the $p$-minimal realisation, it follows that $((G_i, P_i),) \uparrow P_{\text{Loop}} \Rightarrow (O_P \equiv L)$.

## 4. $\Sigma$-normal representation

This section investigates a uniform representation for unobservability maps. As before, the plant is assumed
to be trim and the plant language regular. Specifically, let $G = (Q, \Sigma, \delta, q_0, Q)$ be a trim (i.e., accessible and co-accessible) finite-state automaton model that models the discrete-event dynamics of a physical plant, where $Q = \{q_k : k \in \mathbb{I}_Q\}$ is the set of states. In addition to trimness, we assume that the plant model satisfies the following property

$$\forall q_j \in Q. \exists \sigma \in \Sigma, \text{ such that } \delta(q_j, \sigma) \text{ is defined.}$$ (33) 

The above property ensures that every state has at least one outgoing event defined. Note the outgoing event can be a self loop. It follows that this can be easily guaranteed by, if necessary, augmenting the alphabet as

$$\Sigma_{\text{aug}} = \Sigma \cup \{\$\} \quad \text{(Property A)}$$

where $\$ is a symbol not in $\Sigma$ and adding self-loops to states which lack any outgoing event as illustrated in Figure 12. We note that the above modification may be necessary only for partial or incomplete automata.

**Definition 19:** For an event $\sigma \in \Sigma$, the relative history ($\mathcal{H}_\sigma$) is defined as:

$$\mathcal{H}_\sigma = \{\omega \in L_m(G_i) : \delta(\delta^*(q_i, \omega), \sigma) \in Q\}. \quad (34)$$

**Definition 20:** For an event $\sigma \in \Sigma$, and a given state $q_k \in Q$, the relative history at state $q_k$ ($\mathcal{H}_{\sigma|q_k}$) is defined as:

$$\mathcal{H}_{\sigma|q_k} = \{\omega \in L_m(G_i) : \delta(\delta^*(q_i, \omega), \sigma) \in Q\}. \quad (35a)$$

The following points are worth noting.

- We have

$$L_m(G_i) = \bigcup_{\sigma \in \Sigma} \mathcal{H}_\sigma. \quad (35b)$$

The second equality is valid for trim models that satisfy the Property A described above and is necessary for the development in the sequel.

- For $\sigma_1, \sigma_2 \in \Sigma$, $\mathcal{H}_{\sigma_1|q_k}$ and $\mathcal{H}_{\sigma_2|q_k}$ are not necessarily disjoint. In fact, it follows that if both $\sigma_1$ and $\sigma_2$ are defined from the state $q_k$, we have $\mathcal{H}_{\sigma_1|q_k} \cap \mathcal{H}_{\sigma_2|q_k} \equiv L(q_i, q_k)$ and if there is no $\sigma_1$ defined from $q_k$ then $\mathcal{H}_{\sigma_1|q_k} = \emptyset$.

Thus, $\mathcal{H}_{\sigma_1|q_k} \in \{\emptyset, L(q_i, q_k)\}$. 

**Definition 21:** For a given alphabet $\Sigma = \{\sigma_0, \ldots, \sigma_m\}$ and an unobservability map $p$, the $\Sigma$-normal phantom set $L$ is a set of languages ($L_{\sigma_0}, \ldots, L_{\sigma_m}$) such that, $(\forall \sigma_i \in \Sigma) (L_{\sigma_i} \subseteq \mathcal{H}_\sigma) \land ((\forall \omega \in \mathcal{H} \sigma_i(p_\sigma(\sigma_i) = \epsilon) \Rightarrow (\omega \in L_{\sigma_i}))$ 

**Proposition 7 (\Sigma-normal Representation Theorem):** The $\Sigma$-normal phantom set $L$ is uniquely specified by the unobservability map $p$. Conversely, any set of languages ($L_{\sigma_0}, \ldots, L_{\sigma_m}$) which satisfies $L_{\sigma_i} \subseteq \mathcal{H}_\sigma, \forall \sigma_i \in \Sigma$ uniquely corresponds to an unobservability map $p$.

**Proof:** Let $p$ be an unobservability map. If possible let there be two $\Sigma$-normal phantom sets $L^1 = \{L_{\sigma_0}^1, L_{\sigma_1}^1, \ldots\}$ and $L^2 = \{L_{\sigma_0}^2, L_{\sigma_1}^2, \ldots\}$. Now, $\omega \in L^1_{\sigma_i} \Rightarrow p_\sigma(\sigma_i) = \epsilon \Rightarrow \omega \in L^2_{\sigma_i}$. Hence, $L^1_{\sigma_i} \subseteq L^2_{\sigma_i} \forall \sigma_i \in \Sigma$. Similarly, $L^2_{\sigma_i} \subseteq L^1_{\sigma_i} \forall \sigma_i \in \Sigma$. Hence, $L^1_{\sigma_i} = L^2_{\sigma_i}$. For an arbitrary set of languages $\mathcal{L} = \{L_{\sigma_0}, \ldots, L_{\sigma_m}\}$ that satisfies $L_{\sigma_i} \subseteq \mathcal{H}_\sigma, \forall \sigma_i \in \Sigma$, we have,

$$(L_{m}(G_i) = \bigcup_{\sigma \in \Sigma} \mathcal{H}_\sigma) \Rightarrow (L_{m}(G_i) = \bigcup_{\sigma \in \Sigma} (L_{\sigma_i} \cup (\mathcal{H}_\sigma \setminus L_{\sigma_i}))).$$ (36) 

Equation (36) allows us to define a map $\varphi : \Sigma^* \rightarrow \Sigma \cup \{\$\}$ as follows:

$$\forall \omega \in \Sigma^* \text{ with } \omega = \lambda \sigma_i \text{ for some } \sigma_i \in \Sigma, \lambda \in \Sigma^* , \Rightarrow \varphi(\lambda, \sigma_i) = \begin{cases} \sigma_i, & \text{if } \lambda \in (\mathcal{H}_\sigma \setminus L_{\sigma_i}) \\ \epsilon, & \text{if } \lambda \in L_{\sigma_i}. \end{cases}$$

The map $\varphi$ is well-defined because $(\mathcal{H}_\sigma \setminus L_{\sigma_i}) \cap L_{\sigma_i} = \emptyset$. Next we define an induced map $\tilde{\varphi} : \Sigma^* \rightarrow \Sigma^*$ as follows:

$$\forall \omega = \sigma_1 \cdots \sigma_j \cdots \sigma_m, \text{ with } \sigma_j \in \Sigma, \Rightarrow \tilde{\varphi}(\omega) = \varphi(\epsilon, \sigma_1) \cdots \varphi(\sigma_1 \cdots \sigma_{j-1}, \sigma_j) \cdots \varphi(\sigma_1 \cdots \sigma_{m-1}, \sigma_m).$$

$\tilde{\varphi}$ satisfies the basic properties of an unobservability map, namely, $\tilde{\varphi}(\epsilon) \in \{\epsilon, \sigma\}$ by definition of $\varphi$ and $\tilde{\varphi}(\omega \sigma) = \tilde{\varphi}(\omega) \varphi(\sigma)$ if $\tilde{\varphi}(\omega) \sigma, \tilde{\varphi}(\omega) \sigma \in \mathcal{H}_\sigma$. Hence, there exists at least one unobservability map $p \equiv \tilde{\varphi}$ which corresponds to $\mathcal{L}$. Let, if possible, $p^1$ and $p^2$ be two such maps with $p^1 \neq p^2$. Now, $(p^1 \neq p^2) \Rightarrow (3 \omega \in \Sigma^*, \sigma \in \Sigma, \text{ such that } p^1_\sigma(\sigma) \neq p^2_\sigma(\sigma))$. Without loss of generality, it can be assumed that $p^1_\sigma(\epsilon) = \epsilon$ and $p^2_\sigma(\epsilon) = \sigma$. Now, $(p^1_\sigma(\epsilon) = \epsilon) \Rightarrow (\omega \in L_{\sigma_i})$ and $(p^2_\sigma(\epsilon) = \sigma) \Rightarrow (\omega \in (\mathcal{H}_\sigma \setminus L_{\sigma_i}))$. But $\forall \sigma_i \in \Sigma((\mathcal{H}_\sigma \setminus L_{\sigma_i}) \cap L_{\sigma_i} = \emptyset)$
which implies $p^1$ and $p^2$ coincide everywhere (contradiction). Hence, $p$ is unique.

Remark 6: It follows from Proposition 7 that the $\Sigma$-normal representation of the unobservability map $p$. In section 3, it was proved that the $p$-minimal realisation is unique. However, the normal form of Proposition 7 has the advantage that it admits a finite representation for a much wider class of unobservability maps. For example, it will be proved shortly that if any element of the phantom set is just context free non-regular and the rest are all regular, the $p$-minimal realisation is infinite, whereas there exists many finite descriptions for context free languages (e.g. context free grammars) and hence a finite normal form for $p$ exists. The same argument works as long as all elements of the phantom set are recursively enumerable.

Example 3: An example illustrates the idea. Figure 13 represents a trim DFSA with $L_m(G_1) = \{0, 1\}^*$ with the following property:

The unobservability map $p$ is such that the first 0 generated is unobservable and so is every alternate 0 after that.

The $p$-minimal realisation is given in Figure 14. The $\Sigma$-normal phantom set in this case is as follows:

$$L = \{L_0, L_1\} \text{ where } \begin{cases} L_0 = 1^*(01^*01^*)^* \\ L_1 = \emptyset. \end{cases}$$

$L_1$ is empty since 1 is always observable. $L_0$ is the language of all strings having an even number of 0s and hence the 0 generated after a history belonging to $1^*(01^*01^*)^*$ will be unobservable according to the map $p$. This applies to the first 0 generated, since $1^* \subset 1^*(01^*01^*)^* \equiv L_0$. Note that the $p$-minimal realisation is finite and every element of $L$ is regular. It will be shown in the sequel that this is a general result.

The following properties follow immediately from definition:

- For a perfectly observable system, the $\Sigma$-normal phantom set consists of $\ell$ copies of the null set, where $\ell$ is the cardinality of the alphabet, i.e., $p = \emptyset$ if and only if $L = \{\emptyset, \ldots, \emptyset\}$. This follows from the fact that if an event $\sigma$ is always observable, irrespective of its past, then $L_\sigma = \emptyset$.

- For a completely unobservable system, $L = \{\mathcal{H}_\sigma : \sigma \in \Sigma\}$. This follows from the fact that if an event $\sigma$ is always unobservable, then the set of histories for which $\sigma$ is unobservable is, in fact, the entire set of histories for the event $\sigma$ by definition.

- State independent unobservability can be identified with the following case: $L = \{L_\sigma : L_\sigma = \bigcup_{i \in B_\sigma} \mathcal{H}_\sigma q_i\}$, where $B_\sigma$ is some arbitrary subset of $L_\sigma$.

- State dependent unobservability can be identified with the following case: $L = \{L_\sigma : L_\sigma = \bigcup_{i \in B_\sigma} \mathcal{H}_\sigma q_i\}$, where $B_\sigma$ is some arbitrary subset of $L_\sigma$.

Next we prove that the $p$-minimal realisation is finite if and only if all elements of the $\sigma$-normal phantom set are regular.

Proposition 8: $\forall \sigma \in \Sigma, (L_\sigma \in \text{REG}) \iff (G_\sigma, p_\sigma), \downarrow$.

Proof: First we show, $(G_\sigma, p_\sigma), \uparrow \Rightarrow \exists \sigma \in \Sigma$, such that $L_\sigma \notin \text{REG}$. Assume $(G_\sigma, p_\sigma), \downarrow$. This implies that the projective Nerode equivalence relation is not of finite index. Hence it follows that there exists $H = \{h_1, h_2, \ldots, h_r\}$ with $r \in \mathbb{N}$ such that $\forall h_r, h_j \in H, \exists u \in \Sigma^*$ such that $p_\sigma^h(u) \neq p_\sigma^h(u)$. Now, finiteness of $\Sigma$ implies that for some $\sigma \in \Sigma$, there exists an infinite set $H_\sigma \subseteq H$ such that $\forall h_r, h_j \in H_\sigma, \exists u \in \Sigma^*$ with $p_\sigma^h(u) \neq p_\sigma^h(u)$. Denote the Nerode equivalence on $\Sigma^*$ induced by $L_\sigma$ by $N_{L_\sigma}$. We note that $x N_{L_\sigma} y \Rightarrow \forall u \in \Sigma^* (x u \in L_\sigma \Rightarrow y u \in L_\sigma)$. Let $h_i, h_j \in H_\sigma$. Then we have $h_i N_{L_\sigma} h_j$, but for some $u \in \Sigma^*$, $h_i N_{L_\sigma} h_j u$. Infinite cardinality of $H_\sigma$ implies $N_{L_\sigma}$ is of infinite index. Hence $L_\sigma \notin \text{REG}$. The converse follows immediately by noting that if $(G_\sigma, p_\sigma), \downarrow$, then a finite state description for $L_\sigma$ is obtained by defining as accepting states all states (of $(G_\sigma, p_\sigma)$) at which the event $\sigma, 0$ is defined and replacing the labels $(\sigma, b)$ with $\sigma$. (Note $b \in \{0, 1\}$). This completes the proof.

---

**Figure 13.** Trim plant model.

**Figure 14.** $p$-minimal realisation: Unobservable transition in dashed line.
Remark 7: Proposition 8 demonstrates that the p-minimal realisation and the Σ-normal description of unobservability maps are equivalent.

A classification of the unobservability maps in the Chomsky sense can be done as shown in Table I. It is to be noted that the plant language is regular in all cases.

5. Decidability of state determinacy problem

For the purpose of this section, we consider a trim deterministic finite state plant $G = (Q, \Sigma, \delta, q_0, Q_m)$ with $\Sigma = \{\sigma_0, \ldots, \sigma_{m-1}\}$ and a specified Σ-normal phantom set $L = \{e_0, \ldots, e_{m-1}\}$. Note that determinism implies that for a given generated event sequence, the current plant state is unique. However, given an observed event sequence, the current plant state, in general, belongs to a possible set of states.

Definition 22: The instantaneous description $\overline{D} : p(E(G_j)) \rightarrow 2^Q$ is a map from the set of observed event traces to the power set of finite state automaton states $Q$, such that given an observed event trace $\omega$, $\overline{D}(\omega) \subseteq Q$ is the set of states that the underlying deterministic finite state plant can possibly occupy at the given instant.

Note, in the case of perfect observation, the instantaneous description for any observed sequence is a singleton state set.

Definition 23: Given an observed event sequence $\omega$ and a fixed state $q_j \in Q$, the state determinacy problem is the determination of the truth value of the following logical sentence:

$$D(q_j, \omega)(q_j \in \overline{D}(\omega)). \quad (37)$$

That is, given an observed event sequence, the state determinacy problem is computation of the possible set of current states. The simpler problem where the observed sequence is the empty string is

$$D(q_j, \varepsilon) := D(q_j) := (q_j \in \overline{D}(\varepsilon)). \quad (38)$$

In the sequel, we denote the former problem (Equation 37) by $DG$ and the latter by $DE$ (Equation 38).

Before we investigate decidability of $DE$ and $DG$, we need the concept of phantom reachability trees.

5.1. Construction of phantom reachability trees

Algorithm 1: Derivation of Phantom Reachability Tree $T(q_j)$

Input: $G, L, q_j$
Output: $T(q_j)$

begin
2. Define the root node and label the root as $q_j$ /* Initiate Tree Construction */
3. For each new node $q_k$ in the frontier of the current tree do
4. Compute $A = \text{CHILD}(q_k)$; /* Compute possible child nodes */
5. For each $(q_k, \sigma) \in A$
6. If $q_k$ has an ancestor $q_j$ then
7. Delete $(q_k, \sigma)$ from $A$; /* Avoid repetition in the same ancestry (See Figure 15) */
8. endif
9. endfor
10. For each $(q_k, \sigma) \in A$ do
11. Compute the path $\omega \in \Sigma^*$ from the root to $q_k$.
12. Compute $\text{DEL_NODE}(q_k, \omega)$;
13. If $\text{DEL_NODE}(q_k, \omega) \in \text{\Omega}$ then
14. Delete $(q_k, \sigma)$ from $A$; /* Transitivity check */
15. endif
16. For each remaining $(q_k, \sigma) \in A$ do
17. Create a node labelled $q_k$ connected by an edge labelled $\Sigma$ to $q_k$;
18. endfor
19. endfor
20. Terminate if no new child nodes can be created;
21. endfor
22. end

Figure 15. Tree: $q_j$ is an ancestor of $q_3$; $q_6$ is not.

Definition 24: Given a DFSA plant model $G_0 = (Q, \Sigma, \delta, q_0, Q_m)$ and a Σ-normal phantom set $L$, we define the following:

- $\text{CHILD} : Q \rightarrow 2^Q \times \Sigma$ such that $\forall q_k, q_j \in Q, \sigma \in \Sigma,$
  \[(q_k, \sigma) \in \text{CHILD}(q_j) \iff \exists \sigma \in \Sigma | \delta(q_k, \sigma) = q_j \land q_k \neq q_j. \quad (39)\]
Proposition 9:

1. \( \forall q_j \in Q, \mathcal{F}(q_j) \) is finite.
2. \( q_0 \in \mathcal{F}(q_j) \iff q_j \in \overline{Q}() \). (Note, \( q_0 \) is the initial state of the underlying plant \( G_0 = (Q, \Sigma, \delta, q_0, Q_m) \).)

Proof: For statement 1, we note that by Line 7 of Algorithm 1, any path from the root to the tree front has length bounded by \( \text{Card}(Q) \) (since each node label occurs at most once in every path from the root). Also, each node has at most \( \text{Card}(E) \) immediate children; implying \( \mathcal{F}(q_j) \) is finite for any \( q_j \in Q \). For statement 2, we note that if \( q_1 \xrightarrow{\sigma_1} q_2 \xrightarrow{\sigma_2} q_3 \), then it follows from Definition 19 that \( H_{\sigma_1|q_1} \subseteq H_{\sigma_2|q_2} \). Using this fact, we note that \( \sigma_1 \ldots \sigma_m \) is a path in the tree from the root labelled \( q_0 \) to a node labelled \( q_0 \), then we have by construction: \( (H_{\sigma_1|q_0} \sigma_2 \ldots \sigma_{m-1} \ldots \sigma_{2} ) \cap L_{\sigma_1} \cap (\cap_{i=2}^{m-2} L_{\sigma_i} \sigma_{i-1} \ldots \sigma_2 ) \neq \emptyset \iff \exists q_1, \ldots, q_{m-2} \in Q \) such that \( \sigma_1 \ldots \sigma_{m-1} \sigma_2 \) is a path in the tree from the root labelled \( q_0 \) to a node labelled \( q_0 \), then we have by construction: \( (H_{\sigma_1|q_0} \sigma_2 \ldots \sigma_{m-1} \ldots \sigma_{2} ) \cap L_{\sigma_1} \cap (\cap_{i=2}^{m-2} L_{\sigma_i} \sigma_{i-1} \ldots \sigma_2 ) \neq \emptyset \iff \exists h \in L(G_0) \) such that \( h \in L(q_0, q_0) \) and \( q_0 \xrightarrow{\sigma_1} q_1 \xrightarrow{\sigma_2} q_2 \xrightarrow{\sigma_3} \ldots q_{m-2} \xrightarrow{\sigma_{m-1}} q \) with \( p_\delta(\sigma_m) = \varepsilon \). Hence \( \mathcal{F}(q_j) \) has a node labelled \( q_0 \) if it exists from a completely unobservable path \( \sigma_{m} \ldots \sigma_1 \) from \( q_0 \) to \( q_j \), i.e. \( q_0 \in \overline{Q}() \).}

Proposition 10: Given a DFSA plant model and a \( \Sigma \)-normal phantom set \( L \), \( \mathcal{D}_E \) is decidable if \( L \subseteq \mathcal{L} \) and \( R \in \text{REG}((L_1 \cap L_2 = \emptyset \iff R = \emptyset) \text{ and } (\forall \omega \in \Sigma^*, L_1 \omega \in \mathcal{L}) \), i.e. \( \mathcal{D}_E \) is decidable if the \( \Sigma \)-normal phantom set belongs to a family of languages \( \mathcal{L} \) which is closed under concatenation with singleton strings and for which the emptiness-checking problem for intersection of elements in \( \mathcal{L} \) and for intersection of elements in \( \mathcal{L} \) with regular languages can be effectively solved.

Proof: The result follows immediately by noting that these three properties guarantee that \( \text{Del}_N(q_j, \omega) \) can be effectively computed, and hence the phantom tree \( \mathcal{F}(q_j) \) can be constructed algorithmically for any \( q_j \in Q \).

Proposition 11: Let \( G_0 \) be an arbitrary DFSA model and \( L \subseteq \mathcal{L} \) be the \( \Sigma \)-normal phantom set such that for the language-family \( \mathcal{L} \), the emptiness-checking problem for intersection of two arbitrary languages is undecidable. Then \( \mathcal{D}_E \) is undecidable for this class of \( L \).

Proof: Assume there exists an algorithm for \( \mathcal{A} \) for solving \( \mathcal{D}_E \) for any arbitrary plant model such that the \( \Sigma \)-normal phantom set \( L \subseteq \mathcal{L} \). Further, we know that for two arbitrary languages \( L_1, L_0 \in \mathcal{L} \), \( L_1 \cap L_0 = \emptyset \iff \exists \emptyset \). We choose the plant model \( G_0 = (\{q_0, q_1, q_2, \{0, 1\}, \delta, q_0, \{q_2\}) \) as shown in Figure 16. Let \( L_1, L_0 \in \mathcal{L} \). We define \( L_1 = L_1 \) and \( L_0 = L_0 \). Now assume the \( \Sigma \)-normal phantom map satisfies the following criteria:

\[
(\mathcal{L}_1 \cap \mathcal{L}_2|_{q_1} = \emptyset) \wedge (\mathcal{L}_1 \cap \mathcal{L}_2|_{q_2} = \emptyset) \\
\wedge (\mathcal{L}_0 \cap \mathcal{L}_2|_{q_0} = \emptyset) \wedge (\mathcal{L}_0 \cap \mathcal{L}_1|_{q_0} = \emptyset) \\
\wedge (\mathcal{L}_1 = L_1) \wedge (\mathcal{L}_0 = L_0).
\]

Since \( \mathcal{L} \subseteq \mathcal{L} \), we use algorithm \( \mathcal{A} \) to answer \( q_2 \in \overline{Q}() \). We note that \( q_2 \in \overline{Q}() \iff (\mathcal{L}_1 \cap \mathcal{L}_2|_{q_1} = \emptyset) \wedge (\mathcal{L}_0 \cap \mathcal{L}_2|_{q_0} = \emptyset) \wedge (\mathcal{L}_0 \cap \mathcal{L}_1|_{q_0} = \emptyset) \wedge (\mathcal{L}_1 = L_1) \wedge (\mathcal{L}_0 = L_0).
\]

Since \( (\mathcal{L}_1 \cap \mathcal{L}_2|_{q_1} = \emptyset) \wedge (\mathcal{L}_1 \cap \mathcal{L}_2|_{q_2} = \emptyset) \), it follows that \( \mathcal{L}_1 \subseteq \mathcal{L}_2 \), implying

\[
(\mathcal{L}_1 \cap \mathcal{L}_2) \cap \mathcal{L}_2|_{q_1} = \emptyset \iff (\mathcal{L}_1 \cap \mathcal{L}_0) = \emptyset \\
\iff (\mathcal{L}_1 \cap \mathcal{L}_0) = \emptyset \iff (\mathcal{L}_1 \cap \mathcal{L}_2) = \emptyset \iff (\mathcal{L}_1 \cap \mathcal{L}_2) = \emptyset \iff \emptyset \iff \emptyset.
\]

Thus \( \mathcal{A} \) allows us to decide the emptiness-check for intersection of arbitrary languages in \( \mathcal{L} \) implying no such algorithm exists.

Corollary 1: \( \mathcal{D}_E \) and \( \mathcal{D}_G \) are undecidable for the class of problems where \( L \) is a set of arbitrary context-free languages.

Proof: For two arbitrary context-free languages \( L_1, L_2, L_1 \cap L_2 = \emptyset \) is undecidable (Ramadge and Wonham 1987; Ray 2005; Chattopadhyay and...
Ray 2007a). Hence it follows immediately from Proposition 11 that DE is undecidable in this case. We note that “DG is decidable” implies there exits an algorithm that answers $q_j \in Q(\omega)$ for all $\omega \in \Sigma$. Hence one can use the same algorithm to answer $q_j \in Q(\epsilon)$ i.e. “DG is decidable” $\Rightarrow$ “DE is decidable”. Hence, by contrapositive, “DE is not decidable” $\Rightarrow$ “DG is not decidable”. This completes the proof.  

5.2. State determinacy for regular unobservability

It follows from Proposition 10 that DE is decidable and in fact computable in polynomial time if $l \subset \text{REG}$. We show that for regular unobservability DG is decidable as well. In case of regular unobservability, the result of Proposition 10 can be applied to construct a Petri net observer (Moody and Antsaklis 1998). The advantage of using a Petri net description is the compactness of representation and simplicity of the online execution algorithm that we present next. We are interested in computing the set of states that the plant may possibly be in, given an observed sequence of events. Our preference of a Petri net description over a subset construction for finite state machines is motivated by the following:

Algorithm 2: Petri net observer for reg. unobservability

```
Input: (G, p)
Output: Petri net observer

1. begin
2. I. Create a place $q_i$ for each state $q_i$ in (G, p);
3. II. The set of transition labels is $\Sigma$;
4. For each observable transition $q_j \rightarrow q_k$ in (G, p) do
5. I. Set the initial state in (G, p) to $q_i$;
6. II. Compute $Q(\epsilon)$;
7. III. Add a transition labelled $\sigma$ from the place $q_i$ with output arcs to all places $q_j \in Q(\epsilon)$;
8. endfor
9. For each place $q_j$ in the net do
10. For each event $\sigma \in \Sigma$ do
11. If there is no transition with label $\sigma$ from $q_j$ then
12. I. Add a flush-out arc with label $\sigma$ from $q_j$
13. endif
14. endwhile
15. endwhile
16. end
```

1. The Petri net formalism is natural, due to its ability to model transitions of the type $q_1 \rightarrow_{\epsilon} q_3$, which reflects the condition “the plant can possibly be in states $q_2$ or $q_3$ after an observed transition from $q_1$”.
2. One can avoid introducing an exponentially large number of “combined states” of the form $[q_2, q_3]$ as involved in the subset construction and more importantly preserve the state description of the underlying plant.

3. This lays the groundwork for future extension to handle probabilistic models when occupancy distributions over possibly occupied states can be obtained i.e. make conclusions such as “the current state is $q_2$ with probability 0.6 and $q_3$ with probability 0.4” as opposed to “the plant is now in the combined state $[q_2, q_3]$”.

Since for regular unobservability, the $p$-minimal realisation is finite, we can construct a Petri net with flush-out arcs as given by Algorithm 2. Flush-out arcs were introduced by Gribaudo et al. (2001) in the context of fluid stochastic Petri nets. We apply this notion to ordinary nets with similar meaning: a flush-out arc is connected to a labelled transition, which, on firing, removes a token from the input place (if the arc weight is one).

**Proposition 10:** For regular unobservability,

1. Algorithm 2 has polynomial complexity.
2. Once the Petri net observer has been computed off line, the current possible states for any observed sequence can be executed by computing Algorithm 3 online:

Algorithm 3: Online computation of possible states

```
input : Petri net observer, Observed sequence $\omega = \tau_1, \tau_2, \ldots, \tau_r$
output: $Q(\omega)$
1. begin
2. I. Compute the initial marking for the observer as follows:
3. a. Compute $Q(\epsilon)$;
4. b. Put a token in each place $q_j \in Q(\epsilon)$;
5. For $j = 1$ to $r$ do
6. I. Fire all enabled transitions labelled as $\tau_j$ in the observer;
7. For each place $q_j$ in the observer do
8. if number of tokens in $q_j > 0$ then
9. I. Normalise the number of tokens in $q_j$ to 1.
10. endif
11. endif
12. endfor
13. II. $Q(\omega) = \{q_j \mid q_j \text{ has one token}\}$
14. end
```

**Proof:** The complexity claim (Assertion 1) follows immediately from noting that the only non-trivial step is the computation of $Q(\epsilon)$ which involves performing the emptiness-check for arbitrary regular languages which can be executed in polynomial time (Ramadge and Wonham 1987; Ray 2005; Chattopadhyay and Ray 2007a). For Assertion 2, we note that the computation of the initial marking for the observation net follows from the definition of $Q(\epsilon)$ (See Definition 22). If an event $\Sigma$ is observed, Algorithm 2 implies that firing all $\sigma$ labelled transitions in the
observer is equivalent to computing the set of states that the plant may reach from each possible current state. Furthermore, if $q_i$ is marked initially and $\sigma$ is not defined from $q_i$ in the underlying plant, then observing $\sigma$ implies that the plant was not in $q_i$. This is taken care of by the $\sigma$-labelled $\text{flush-out arc}$ in the observer (exists by construction), which flushes out or eliminates tokens from the place $q_i$. We only want to tag places as “possibly occupied” or “not occupied” and do not want token accumulation; hence the normalisation step in Algorithm 3. Note that the firing sequence of the enabled transitions is not important; however it is important to only fire those transitions that were enabled before initiating the firing sequence at each step.

6. An example

This section presents a game between two players PATROL (denoted by $\Box$) and INTRUDER (denoted by $\Diamond$) as an example.

Remark 7: Having two players in the game suffices to illustrate the underlying concepts and algorithms without loss of generality, because the critical issue here is complexity of the unobservability maps with respect to the individual players instead of the number of players. The algorithms described in the previous sections pertain to individual agents and their corresponding unobservability maps; the algorithmic complexities (either explicit run-times or their asymptotic estimates) have no dependence on the number of agents involved in any particular situation.

The game is played in three rooms as shown in Figure 17. Rooms R1 and R2 are homes for PATROL and INTRUDER respectively. The objective for INTRUDER is to remove targets (shown as black dots) from room $H$ without being intercepted by PATROL. If INTRUDER operates in stealth mode, PATROL cannot detect his entrance in room $H$. However, it is costly for INTRUDER to engage stealth mode. Further, it is costly for PATROL to wait in room $H$; it is cheaper for her to retreat to home. Also, PATROL can always detect INTRUDER entering $H$ if she already is in $H$. Both players must attempt to achieve respective objectives preferably at the minimum mission cost. We explore strategies for PATROL for different strategies employed by INTRUDER. First, assume INTRUDER chooses to operate always under stealth (bearing a high mission cost). The $p$-minimal realisation of the plant with the unobservable transition in a dashed line is shown on the left side of Figure 19. The interpretation of the states and events are shown in Table 2. The unobservability map is regular; the $p$-minimal realisation is finite and for the ordered alphabet $\Sigma = \{e, r, a\}$, the $\Sigma$-normal phantom set is given by $L = \{L_1 = \{\text{all strings terminating at state 00}, \emptyset, \Diamond\}$. (Note that all elements are regular.). PATROL applies Algorithm 2 to construct the Petri net observer as shown on the right of Figure 19. The flush-out arcs are in dotted. Assuming 00 to be the initial state in the underlying plant, the initial marking in the observer is computed to be $\{1,1,0,0\}$ with the places ordered as 00, 01, 11, 10. PATROL can now keep track of the possible states by firing observed events in the constructed Petri net, e.g.

$$\{1,1,0,0\} \rightarrow \{1,1,0,0\} \rightarrow \{0,0,1,0\} \rightarrow \{1,1,0,0\} \rightarrow \cdots$$

(43)
and take appropriate actions. The first few possibilities are shown in Figure 18. A reachability analysis of the Petri net observer shows that neither of the states \{1, 0, –, –\} and \{0, 1, –, –\} is reachable, which concurs with the situation that PATROL must always wait in \(H\) (i.e., can never detect INTRUDER from home) or risk losing the targets (and hence the game). Next let us assume that INTRUDER chooses to employ a cheaper strategy, engaging stealth mode at every even chance. The unobservability map for PATROL is still regular and the \(p\)-minimal realisation is shown in Figure 20.

Note, that the state 01 has been split into 01 and 01. The phantom set in this case is given by \(L = \{L_2, \emptyset, \emptyset\}\) where \(L_2 = \{\text{all strings terminating at state 01}\}\). Note that \(L_2\) is a regular subset of \(L_1\) as described for the previous situation, computes the observer, shown in Figure 21, and proceeds as before. However, in the current situation, PATROL can reduce her mission cost by selectively retreating to home. If INTRUDER chooses to employ a stealth strategy that makes some elements of the phantom set context-free, PATROL is no longer able to effectively track INTRUDER due to the undecidability result of Proposition 11. Note that in each case, PATROL is aware of INTRUDER’s strategy in the form of the unobservability map, however, the usability of that information is dependent on the complexity of the unobservability situation.

7. Summary, conclusions and future work

This paper addresses the problem of partial observation in discrete-event supervisory systems, specifically attempting to relax the assumption that unobservability only introduces bounded memory in the observed process. A formal framework has been established for generalised projections in DES and theoretical results have been presented that are necessary for further research in the area. It has been shown that the unobservability situations analysed in the reported literature form a special case, namely that of state-dependent and state independent regular unobservability. Furthermore, the normal representation of unobservability maps, introduced in this paper, allow for finite and compact representations of projections that introduce unbounded memory in the observed plant. The problem of observation based estimation of the possible set of current states in a finite state plant is shown to be solvable in polynomial time for regular unobservability maps and undecidable for more complex unobservability situations.

The work reported in this paper raises the following issues that future work should address.

- How does a more complex underlying plant model affect the state determinacy problem? In particular, is the state determinacy problem decidable for non-regular plants with regular unobservability maps?
- Is the state determinacy analysis extensible to probabilistic underlying models? In particular, does the negative decidability result change for context-free unobservability if the underlying plant is a probabilistic finite state language generator?

The first of the above two issues is critical for modelling and analysis of plants that cannot be adequately represented by finite state systems. Examples of such infinite-memory models (e.g. Petri nets) are plentiful in the literature. Positive decidability for Petri nets with unobservable transitions could be very useful for controller design, where the
unobservability situation is simple (e.g., regular). On the other hand, negative decidability is likely to make the task of controller design more complex and may require sensor redundancy.

The second of the above two issues is important for extension of the results, presented in this paper, to plants that have event probabilities associated with state transitions. Since probabilistic finite state machines are closely related to finite state Markov chains, extension of the current analysis to probabilistic models may lead to valuable insights in controlling partially observable Markov decision processes (POMDP).

Acknowledgements
This work has been supported in part by the U.S. Army Research Laboratory and the U.S. Army Research Office under Grant Nos. W911NF-07-1-0376 and W911NF-06-1-0469.

References
Wonham, W. (2001), Control of Discrete-event Systems, Department of Electrical Engineering, University of Toronto, Ontario, Canada.