International Journal of Control
Vol. 81, No. 10, October 2008, 1626–1644

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 Σ-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

*Corresponding author. Email: axr2@psu.edu

ISSN 0020–7179 print/ISSN 1366–5820 online

DOI: 10.1080/00207170801888837
http://www.informaworld.com

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 Σ-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 $\wedge$ and $\vee$, 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 cardinality of the state set Q respectively; **REG** denotes the set of regular languages; $|$ is used interchangeably with the expression "such that"; $|\omega|$ denotes the (non-negative integer) length of the event string ω; and $\mathbb{N}$ is the set of natural numbers. If $\mathcal{N}$ is an equivalence relation, then $x\mathcal{N}y$ implies that x and y belong to the same equivalence class of $\mathcal{N}$, while $x \not\mathcal{N} y$ implies that they belong to distinct equivalence classes.

A trim (i.e., accessible and co-accessible) deterministic finite-state automaton (DFSA) $G_{\mathrm{i}} = \langle Q, \Sigma, \delta, q_{\mathrm{i}}, Q_m\rangle$ represents the discrete-event dynamics of a physical plant, where $Q = \{q_k : k \in \mathcal{I}_Q\}$ is the set of states and $\mathcal{I}_Q \equiv \{1, 2, \ldots, n\}$ is the index set of states; the automaton starts with the initial state $\mathrm{q_i}$; the alphabet of events is $\Sigma = \{\sigma_k : k \in \mathcal{I}_\Sigma\}$, having $\Sigma \cap \mathcal{I}_{\mathrm{Q}} = \ominus$ and $\mathcal{I}_\Sigma \equiv \{1, 2, \ldots, \ell\}$ is the index set of events; $\delta : Q \times \Sigma \to Q$ is the (possibly partial) function of state transitions; and $Q_m \equiv \{q_{m_1}, q_{m_2}, \ldots, q_{m_l}\} \subseteq Q$ is the set of marked (i.e., accepting) states with $q_{m_k} = q_j$ for some $j \in \mathcal{I}_Q$. Let Σ^* be the Kleene closure of Σ, i.e., the set of all finite-length strings made of the events belonging to Σ as well as the empty string ϵ that is viewed as the identity of the monoid Σ^* under the operation of string concatenation, i.e., $\epsilon s = s = s\epsilon$. The extension $\delta^* : Q \times \Sigma^* \to Q$ 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 $\overline{L}$.

2.2 *Partial and total automata models*

Partial DFSA refers to models $G_i = (Q, \Sigma, \delta, q_i, Q_m)$ for which the transition function δ is partial as opposed to complete models for which δ is a total function. In the paradigm of discrete event control, partial automata are often used to model plant dynamics (Rmadge 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 (Rmadge 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 $G_i = (Q, \Sigma, \delta, q_i, Q_m)$ is defined to be $L_m(G_i) = s \in \Sigma^* | \delta^*(q_i, s) \in Q_m\}$. $L_m(G_{\mathrm{i}})$ is also referred to as the marked language.

Definition 2: The (regular) language generated by a DFSA $G_i = (Q, \Sigma, \delta, \mathrm{q}_i, Q_m)$ 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 $\overline{L_m(G_i)}$. Note that if the transition function δ is total, then $\overline{L_m(G_i)} = \Sigma^*$. Also, if all states are marked, i.e., if $Q = Q_m$ then $L_m(G_i) = \overline{L_m(G_i)}$.

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

3. Minimal realisation of unobservability maps

In Ramadge and Wonham's original treatment of unobservability (Rmadge and Wonham 1987), the symbol alphabet Σ is partitioned as $\Sigma_o \cup \Sigma_{uo}$ where events in Σ_o are observable from all states at which they are defined while the events in Σ_{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 $p_i : \overline{L_m(G_i)} \longrightarrow \Sigma^*$ is a mapping that satisfies the following conditions:

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

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 Σ to another string over a possibly distinct alphabet T, 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 $\overline{L_m(G_i)}$ and a given unobservability map p_i, the observed language $\mathbb{O}_p$ is defined as $\{\omega \in \Sigma^*$ such that $\exists\, s \in \overline{L_m(G_i)}$ with $p_i(s) = \omega\}$.

Remark 1: Since unobservability maps are prefix-preserving and generated languages are prefix closed, the observed language $\mathbb{O}_\mathrm{p}$ is prefix closed.

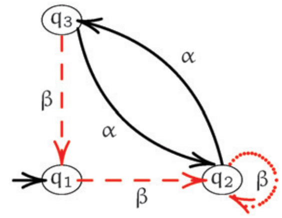

Figure 1. State independent unobservability: observable transitions in solid lines.

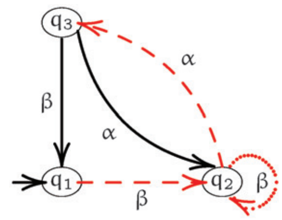

Figure 2. State dependent unobservability: observable transitions in solid lines.

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 β 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 $\overline{L_m(G_i)}$, 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 $\overline{L_m(G_i)}$ 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 λ 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 ω_1 and

ω_2 has no common prefix. Then, prefix subtraction $\vdash: \Sigma^* \times \Sigma^* \longrightarrow \Sigma^*$ is defined as $s_1 \vdash s_2 = \omega_1$.

Definition 7: Let $p_i: \overline{L_m(G_i)} \longrightarrow \Sigma^*$ be an unobservability map for a DFSA G_i. Then, $\forall q_j \in Q$, an induced unobservability map $p_i^{\text{ind}}: L(q_i, q_j) \times \overline{L_m(G_j)} \longrightarrow \Sigma^*$ is defined as $p_i^{\text{ind}}(s, \omega) = \mathfrak{p}_i(s\omega) \vdash p_i(s)$. *Note*: Let $G_j = (Q, \Sigma, \delta, q_j, Q_m)$ be the plant model with initial state set to q_j. We adopt the following notational simplification. Since for all $\omega \in \overline{L_m(G_i)}$, we have $p_i(\omega) = p_i^{\text{ind}}(\epsilon, \omega), p_i^{\text{ind}}(s, \omega)$ is denoted as $p_i^s(\omega)$in the sequel.

Unobservability maps introduced in this paper model the unobservability in the plant dynamics by erasing unobservable transitions from the generated string. The notion is formalised as follows.

Definition 8: (unobservable and observable transitions): Given a DFSA plant G_i with generated language $\overline{L_m(G_i)}$ and an unobservability map p_i, a transition $\sigma \in \Sigma$ with a generated event history $h \in \overline{L_m(G_i)}$ is said to be unobservable if $\mathfrak{p}_i^h(\sigma) = \epsilon$. Likewise, the event $\sigma \in \Sigma$ is said to be observable if $\mathfrak{p}_i^h(\sigma) = \sigma$. We use the following terminology: a string $s \in \overline{L_m(G_i)}$ is called unobservable if at least one of the events in s is unobservable. Similarly, a string $s \in \Sigma^*$ is called completely unobservable if each of the events in s is unobservable.

Definition 9: (state-dependent and state-independent unobservability): A state-independent unobservability map p_i for a DFSA G_i (see Figure 1), is characterised as

$$\forall s, t, s\omega, t\omega \in \overline{L_m(G_i)}(p_i^s(\omega) = p_i^t(\omega)). \tag{1}$$

That is, if an event $\sigma \in \Sigma$ is unobservable from some state $q_k \in Q$, it is unobservable from all states at which it is defined.

A state-dependent unobservability map p_i for a DFSA G_i (see Figure 2), is characterised by the following conditions:

$$G_i \text{ is in its unique minimal realisation} \tag{2a}$$

$$\forall q_j \in Q(\forall s, t \in L(q_i, q_j), \forall \omega \in \overline{L_m(G_j)}, \quad (p_i^s(\omega) = p_i^t(\omega))) \tag{2b}$$

$$\exists q_j, q_k \in Q(\ \exists s \in L(q_i, q_j)\ \exists t \in L(q_i, q_k)\ \exists \omega \in \Sigma^* \text{ such that } (s\omega \in \overline{L_m(G_i)} \bigwedge t\omega \in \overline{L_m(G_i)} \bigwedge p_i^s(\omega) \neq p_i^t(\omega))). \tag{2c}$$

Thus, under state-dependent unobservability, an event $\sigma \in \Sigma$ may be observable at one state $q_j \in Q$ and unobservable at some other state $q_k \in Q$, but the observability of σ is not dependent on how those states are reached.

Given an unobservability map p, the notion of a marked realisation of a DFSA is introduced next. An arbitrary regular language has many automaton representations and a unique minimal representation; a non-minimal representation might be necessary for the following reason : if an event σ_j from a state 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 transitioxsn system (Leeuwen 1990).

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

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

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

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

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

1. there exists a surjective function
$$\text{Merge}: \tilde{Q} \longrightarrow Q \text{ with } \text{Merge}(\tilde{q}_i) = q_i \tag{3a}$$
2. $(\forall \tilde{q}_j, \tilde{q}_k \in \tilde{Q}, \forall \sigma \in \Sigma, (\tilde{q}_j, (\sigma, 0), \tilde{q}_k) \in \tilde{\delta})$
$$\Rightarrow (\forall \tilde{q}_r \in \tilde{Q}, (\tilde{q}_j, (\sigma, 1), \tilde{q}_r) \notin \tilde{\delta}) \tag{3b}$$
3. $(\forall \tilde{q}_j, \tilde{q}_k \in \tilde{Q}, \forall \sigma \in \Sigma, (\tilde{q}_j, (\sigma, 1), \tilde{q}_k) \in \tilde{\delta})$
$$\Rightarrow (\forall \tilde{q}_r \in \tilde{Q}, (\tilde{q}_j, (\sigma, 0), \tilde{q}_r) \notin \tilde{\delta}) \tag{3c}$$
4. $\forall \tilde{q}_j, \tilde{q}_k \in \tilde{Q}, \sigma \in \Sigma$, we have
$$(\tilde{q}_j, (\sigma, 0), \tilde{q}_k) \in \tilde{\delta} \Leftrightarrow \exists h \in \overline{L_m(G_i)}\Big(\big(\mathfrak{p}_i^h(\sigma) = \epsilon\big) \bigwedge (\text{Merge}(\tilde{q}_j) = \delta(q_i, h)) \bigwedge (\text{Merge}(\tilde{q}_k) = \delta(\delta(q_i, h), \sigma))\Big)$$

$$(\tilde{q}_j, (\sigma, 1), \tilde{q}_k) \in \tilde{\delta}$$
$$\Leftrightarrow \exists h \in \overline{L_m(G_i)}\Big(\big(\mathfrak{p}_i^h(\sigma) = \sigma\big) \bigwedge (\textsc{Merge}(\tilde{q}_j)$$
$$= \delta(q_i, h)) \bigwedge (\textsc{Merge}(\tilde{q}_k) = \delta(\delta(q_i, h), \sigma)5) \Big). \tag{3d}$$

The function 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 $\mathfrak{p}_i^s(\sigma) = \epsilon$ and $\mathfrak{p}_i^t(\sigma) = \sigma$ with $\delta(q_i, s) = \delta(q_i, t) = q_j$ in the underlying plant G_i, 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 $\langle G_i, p_i\rangle$. 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 $\langle G_i, p_i\rangle$ with the merging map* Merge, *we have*

$$\exists s, t \in \overline{L_m(G_i)}\Big(\exists \sigma \in \Sigma,\ \mathfrak{p}_i^s(\sigma)$$
$$= \epsilon \bigwedge \mathfrak{p}_i^t(\sigma) = \sigma \bigwedge (\delta(q_i, s) = \delta(q_i, t) = q_j \in Q) \Big)$$
$$\Longrightarrow \mathrm{Card}\big(\textsc{Merge}^{-1}(\{q_j\})\big) > 1. \tag{4}$$

Proof: Denote $\langle G_i, p_i\rangle = (\tilde{Q}, \Sigma \times \{0, 1\}, \tilde{\delta}, \tilde{q}_i)$ and let $\exists s, t \in \overline{L_m(G_i)}(\mathfrak{p}_i^s(\sigma) = \epsilon \bigwedge \mathfrak{p}_i^t(\sigma) = \sigma \bigwedge \delta^*(q_i, s) = \delta^*(q_i, t) = q_j \in Q)$. Since Merge is onto, we have $\forall q_m \in \mathrm{Q}$, $\textsc{Merge}^{-1}(\{q_m\}) \neq \ominus$, i.e., $\mathrm{Card}(\textsc{Merge}^{-1} \times (\{q_j\})) > 0$. Assume if possible $\textsc{Merge}^{-1}(\{q_j\}) = \{\tilde{q}_k\}$. Since $\delta(\delta^*(q_i, s), \sigma)$ is defined, surjectivity of Merge implies that $\exists \tilde{q}_r \in \tilde{Q}$ such that $\textsc{Merge}(\tilde{q}_r) = \delta(\delta(q_i, s), \sigma)$. It follows that $((\mathfrak{p}_i^s(\sigma) = \epsilon) \bigwedge (\textsc{Merge}(\tilde{q}_k) = q_j = \delta^*(q_i,s)) \bigwedge (\textsc{Merge}(\tilde{q}_r) = \delta(\delta^*(q_i, s), \sigma)))$ is true, implying that $(\tilde{q}_k, (\sigma, 0), \tilde{q}_r) \in \tilde{\delta}$ from Property 4 of Definition 11. Similarly, starting with t, we can argue that $\exists q_\ell^{\mathcal{I}} \in \tilde{Q}$ such that $(\tilde{q}_k, (\sigma, 1), q_\ell^{\mathcal{I}}) \in \tilde{\delta}$. This contradicts Property 2 of Definition 11 implying $\mathrm{Card}(\textsc{Merge}^{-1}(\{q_j\})) > 1$. □

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 Σ into unobservable events Σ_u and observable events Σ_o. The observability of an event from a particular state is made dependent on how the state is reached. A particular event σ, 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

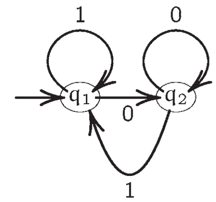

Figure 3. Plant for Example 1.

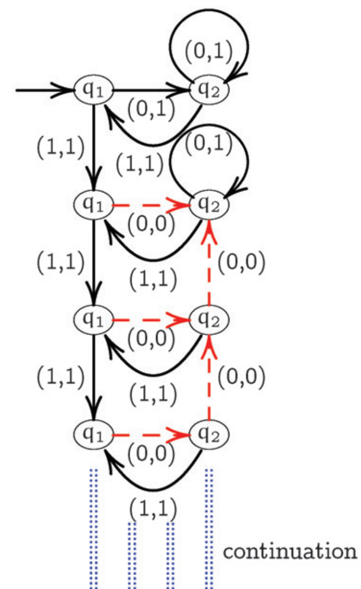

Figure 4. Marked realisation with observable transitions in solid lines.

realisation and there will be some transitions labelled $(\sigma, 0)$ and some as $(\sigma, 1)$ in $\langle G_i, p_i\rangle_*$ all of which correspond to σ (at state q_k) in the underlying plant.

Example 1: It follows from the DFSA in Figure 3 that $\overline{L_m(G_1)} \equiv \{0, 1\}^*$. The unobservability map p 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 $1^n 0^*$ 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 $\mathbb{O}_p$ is regular. In fact, $P: \{0, 1\}^* \leftarrow \{0, 1\}^*$ 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 Σ, the Nerode equivalence relation ($\mathcal{N}$) on Σ^* induced by a language L is defined as $\forall x, y \in \Sigma^*(x\mathcal{N}y \Longleftrightarrow (\forall u \in \Sigma^* \times (xu \in L) \Longleftrightarrow (yu \in L)))$. 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 Σ^* by a given DFSA $G_i = (Q, \Sigma, \delta, q_i, Q_m)$ is the Nerode equivalence on Σ^* induced by the marked or accepted language $L_m(G_i)$ (see section 2C).

Definition 13: Projective Nerode equivalence ($\mathcal{N}_p$) with respect to an unobservability map p on a language L, is defined as $\forall x, y \in \Sigma^*(x\mathcal{N}_p y \Leftrightarrow (x\mathcal{N}y \bigwedge(\forall u \in \Sigma^*(xu \in \overline{L} \Rightarrow (\mathfrak{p}_i^x(u) = \mathfrak{p}_i^y(u))))))$

Lemma 2: *Given a DFSA $G_i = (Q, \Sigma, \delta, q_i, Q_m)$ and an unobservability map $p, \mathcal{N}_p$ is a right invariant equivalence relation on Σ^*, i.e., $\forall x, y, u \in \Sigma^*(x\mathcal{N}_p y \Longrightarrow xu\mathcal{N}_p yu)$.*

Proof: Let $x, y, u \in \Sigma^*$ such that $x\mathcal{N}_p$y. Since, Nerode equivalence $\mathcal{N}$ is right invariant, we have $(x\mathcal{N}_p y) \Rightarrow (x\mathcal{N}y) \Rightarrow (xu\mathcal{N}yu)$. Assume $xu \in \overline{L_m(G_i)}$. Let $v \in \Sigma^*$ such that $xuv \in \overline{L_m(G_i)}$. Now $(x\mathcal{N}_p y)$ implies $\mathfrak{p}_i^x(uv) = \mathfrak{p}_i^y(uv)$ which in turn implies $\mathfrak{p}_i^{xu}(v) = \mathfrak{p}_i^{yu}(v)$. Hence we have $(xu\mathcal{N}yu \bigwedge(\forall v \in \Sigma^*(xuv \in \overline{L_m(G_i)} \Rightarrow (\mathfrak{p}_i^{xu}(v) = \mathfrak{p}_i^{yu}(v))))) \Rightarrow xu\mathcal{N}_p yu$. If $xu \notin \overline{L_m(G_i)}$, then $\nexists v \in \Sigma^* | xuv \in \overline{L_m(G_i)}$ which implies $(\forall v \in \Sigma^*(xuv \in \overline{L_m(G_i)} \Rightarrow (\mathfrak{p}_i^{xu}(v) = \mathfrak{p}_i^{yu}(v))))$ is vacuously satisfied. This completes the proof. □

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

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

$$\forall q \in \tilde{Q} \bigcup\{q_d\}, \tilde{\delta}^*(q, \epsilon) = q \tag{5a}$$

$$\forall \tilde{q}_i, \tilde{q}_k \in \tilde{Q}, \tau \in \Sigma \times \{0,1\},$$

$$\begin{cases} \tilde{\delta}^*(\tilde{q}_i, \tau) = \tilde{q}_k, & \text{iff } \exists \tilde{q}_k \text{ s.t. } (\tilde{q}_i, \tau, \tilde{q}_k) \in \tilde{\delta} \\ \tilde{\delta}^*(\tilde{q}_i, \tau) = q_d, & \text{otherwise} \end{cases} \tag{5b}$$

$$\forall \tau \in \Sigma \times \{0,1\}, \ \tilde{\delta}^*(q_d, \tau) = q_d \tag{5c}$$

$$\begin{aligned} &\forall \tilde{q}_i \in \tilde{Q}, \tau \in \Sigma \times \{0,1\}, \omega \in (\Sigma \times \{0,1\})^*, \\ &\quad \tilde{\delta}^*(\tilde{q}_i, \tau\omega) = \tilde{\delta}^*(\tilde{\delta}^*(\tilde{q}_i, \tau), \omega). \end{aligned} \tag{5d}$$

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 $\tilde{q}_k$ that satisfies $(\tilde{q}_i, \tau, \tilde{q}_k) \in \tilde{\delta}$.

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

$$\phi(\epsilon) = \epsilon \tag{6}$$

$$\phi(\omega\sigma) = \begin{cases} \phi(\omega)(\sigma, 0), & \text{if } \mathfrak{p}_i^\omega(\sigma) \text{ is defined and } \mathfrak{p}_i^\omega(\sigma) = \epsilon \\ \phi(\omega)(\sigma, 1), & \text{otherwise.} \end{cases} \tag{7}$$

Lemma 3: *Following Definition 11, let $\langle G_i, p_i\rangle = (\tilde{Q}, \Sigma \times \{0,1\}, \tilde{\delta}, \tilde{q}_i)$ be a marked realisation. Then*

$$\forall x \in \Sigma^*(\tilde{\delta}^*(\tilde{q}_i, \phi(x)) = \tilde{q}_k \neq q_d \Longrightarrow x \in L(q_i, \textsc{Merge}(\tilde{q}_k)), \tag{8}$$

where q_d and ϕ are given in Definitions 14 *and* 15.

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

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 $\tilde{\delta}^*(\tilde{q}_i, \phi(x\sigma)) = \tilde{q}_\ell \neq q_d$. Then we have $\tilde{\delta}^*(\tilde{\delta}^*(\tilde{q}_i, \phi(x)), \phi(\sigma)) = \tilde{q}_\ell \neq q_d$. Since $\tilde{\delta}^*(q_d, \tau) = q_d$ for all $\tau \in (\Sigma \times \{0,1\})^*$ and $q_\ell \neq q_d$, it follows that

$$\tilde{\delta}^*(\tilde{q}_i, \phi(x)) = \tilde{q}_r \neq q_d \tag{9}$$

$$\Rightarrow \ \tilde{\delta}^*(\tilde{q}_r, \phi(\sigma)) = \tilde{q}_\ell \tag{10}$$

$$\Rightarrow \ \sigma \in L(\textsc{Merge}(\tilde{q}_r), \textsc{Merge}(\tilde{q}_\ell)). \tag{11}$$

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

$$x \in L(q_i, \textsc{Merge}(\tilde{q}_r)). \tag{12}$$

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

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

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

where q_d and ϕ are given in Definitions 14 *and* 15.

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

(*Left to Right*) Assume $x \in \overline{L_m(G_i)}$. 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 MERGE is onto, Definition 11 implies either $(\tilde{q}_i,(x,0),q_\ell^{\mathcal{I}}) \in \tilde{\delta}$ or $(\tilde{q}_i,(x,1),q_\ell^{\mathcal{I}}) \in \tilde{\delta}$ for some $q_\ell^{\mathcal{I}} \in \tilde{Q}$ with $\text{MERGE}(q_\ell^{\mathcal{I}}) = q_k$. In either case, we have $\tilde{\delta}^*(\tilde{q}_i,\phi(x)) = q_\ell^{\mathcal{I}} \neq q_d$.

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 $x \in \overline{L_m(G_i)}$ with $|x|=r \in \mathbb{N}$ such that $\tilde{\delta}^*(\tilde{q}_i,\phi(x)) = \tilde{q}_k \neq q_d$. Then for $\sigma \in \Sigma$ such that $x\sigma \in \overline{L_m(G_i)}$, we have

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

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 $\tilde{\delta}^*(\tilde{q}_i,\phi(x)) = \tilde{q}_k \neq q_d$. It follows from Definition 14 that $(\tilde{q}_i,\phi(x),\tilde{q}_k) \in \tilde{\delta}$ which in turn implies that either $(\tilde{q}_i,(x,0),\tilde{q}_k) \in \tilde{\delta}$ or $(\tilde{q}_i,(x,1),\tilde{q}_k) \in \tilde{\delta}$ and hence $\delta(\text{MERGE}(\tilde{q}_i),x) = \text{MERGE}(\tilde{q}_k)$, i.e., $\delta(q_i,x) = \text{MERGE}(\tilde{q}_k)$ implying $x \in \overline{L_m(G_i)}$. Next we show that if Equation(13) is true from right to left for $|x|=r \in \mathbb{N}$, then it is also true for $x\sigma$ where $\sigma \in \Sigma$. Let $\tilde{\delta}^*(\tilde{q}_i,\phi(x\sigma)) = \tilde{q}_k \neq q_d$. Then we have

$$\tilde{\delta}^*(\tilde{\delta}^*(\tilde{q}_i,\phi(x)),\phi(\sigma)) = \tilde{q}_k \neq q_d \tag{14}$$

$$\Rightarrow \tilde{\delta}^*(\tilde{q}_i,\phi(x)) = \tilde{q}_\ell \neq q_d \quad \text{for some } \tilde{q}_\ell \in \tilde{Q} \tag{15}$$

$$\Rightarrow x \in L(q_i, \text{MERGE}(\tilde{q}_\ell)) \text{ (See Lemma 3).} \tag{16}$$

Also, based on the reasoning for the case $|x|=1$, it follows from Equations (14) and (15) that

$$\tilde{\delta}^*(\tilde{q}_\ell,\phi(\sigma)) = \tilde{q}_k \neq q_d \Rightarrow \sigma \in \overline{L_m(G_\ell)} \quad \text{where } \text{MERGE}(\tilde{q}_\ell) = q_\ell. \tag{17}$$

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

Proposition 1: *Every marked realisation $\langle G_i, p_i\rangle = (\tilde{Q}, \Sigma \times \{0,1\}, \tilde{\delta}, \tilde{q}_i)$ for a given DFSA G_i with a specified unobservability map p_i induces a right invariant equivalence relation $\mathcal{R}_{\langle G_i, pi\rangle}$ on Σ^* defined by*

$$\forall x,y \in \Sigma^*(x\mathcal{R}_{\langle G_i,p_i\rangle}y \Leftrightarrow (\tilde{\delta}^*(\tilde{q}_i,\phi(x)) = \tilde{\delta}^*(\tilde{q}_i,\phi(y)))). \tag{18}$$

Proof: It is readily seen that $\mathcal{R}_{\langle G_i,p_i\rangle}$ is an equivalence relation on Σ^*. For right invariance (Hopcroft et al. 2001), we need to show that

$$\forall \omega \in \Sigma^*, \tilde{\delta}^*(\tilde{q}_i,\phi(x\omega)) = \tilde{\delta}^*(\tilde{q}_i,\phi(y\omega)). \tag{19}$$

For $|\omega|=0$, the result is immediate. Hence we consider $|\omega| \geq 1$. First we note that $\tilde{\delta}^*(\tilde{q}_i,\phi(x)) = \tilde{\delta}^*(\tilde{q}_i,\phi(y))$ implies (see Definition 14) that either $\tilde{\delta}^*(\tilde{q}_i,\phi(x)) = \tilde{\delta}^*(\tilde{q}_i,\phi(y)) = \tilde{q}_k$ for some $\tilde{q}_k \in \tilde{Q}$ or $\tilde{\delta}^*(\tilde{q}_i,\phi(x)) = \tilde{\delta}^*(\tilde{q}_i, \phi(y)) = q_d$. If $\tilde{\delta}^*(\tilde{q}_i,\phi(x)) = \tilde{\delta}^*(\tilde{q}_i,\phi(y)) = q_d$, then Equation (5c) in Definition 14 implies that $\forall \omega \in \Sigma^*, \tilde{\delta}^*(\tilde{q}_i,\phi(x)) = q_d \Longrightarrow \tilde{\delta}^*(\tilde{q}_i,\phi(x\omega)) = q_d = \tilde{\delta}^*(\tilde{q}_i,\phi(y\omega))$. Hence we assume $\tilde{\delta}^*(\tilde{q}_i,\phi(x)) = \tilde{\delta}^*(\tilde{q}_i,\phi(y)) = \tilde{q}_k$ for some $\tilde{q}_k \in \tilde{Q}$. We use induction on the length of ω. Let $|\omega|=1$. Definition 11 implies $\delta(q_i,x) = \text{MERGE}(\tilde{q}_k) = \delta(q_i,y)$ and hence $(x\omega \in \overline{L_m(G_i)} \Leftrightarrow y\omega \in \overline{L_m(G_i)})$. If $x\omega, y\omega \notin \overline{L_m(G_i)}$, then from Lemma 4, we have

$$\tilde{\delta}^*(\tilde{q}_i,\phi(x\omega)) = \tilde{\delta}^*(\tilde{q}_i,\phi(y\omega)) = q_d. \tag{20}$$

Otherwise if $x\omega, y\omega \in \overline{L_m(G_i)}$ we have

$$\delta(q_i,x\omega) = \delta(q_i,y\omega) = \delta(\text{MERGE}(\tilde{q}_k),\omega) = q_\ell \in Q. \tag{21}$$

Now, we claim $\phi(x\omega) = \phi(x)\tau$ and $\phi(y\omega) = \phi(y\tau$ where $\tau \in \{(\omega,0),(\omega,1)\}$. The argument is as follows: Assume if possible $\phi(x\omega) = \phi(x)(\omega,0)$ and $\phi(y\omega) = \phi(y)(\omega,1)$. Since both $\mathfrak{p}_i^x(\omega)$ and $\mathfrak{p}_i^y(\omega)$ are defined (due to $x\omega, y\omega \in \overline{L_m(G_i)}$), we have from Definition 15 that $\mathfrak{p}_i^x(\omega) = \epsilon$ while $\mathfrak{p}_i^y(\omega) = \omega$. It follows that both $(\tilde{q}_k,(\omega,0),\tilde{q}_{\ell 1}) \in \tilde{\delta}$ and $(\tilde{q}_k,(\omega,1),\tilde{q}_{\ell 2}) \in \tilde{\delta}$ for some $\tilde{q}_{\ell 1}, \tilde{q}_{\ell 2} \in \text{MERGE}^{-1}(\{q_\ell\})$ which contradicts Properties 2 and 3 of Definition 11. Note $\tilde{q}_{\ell 1}, \tilde{q}_{\ell 2}$ are not necessarily distinct. Therefore, we have $\phi(x\omega) = \phi(x)\tau$ and $\phi(y\omega) = \phi(y)\tau$ implying that

$$\tilde{\delta}^*(\tilde{q}_i,\phi(x\omega)) = \tilde{\delta}^*(\tilde{q}_k,\tau) = \tilde{\delta}^*(\tilde{q}_i,\phi(y\omega)). \tag{22}$$

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 $\omega\sigma$ where $\sigma \in \Sigma$. Let $|\omega|=r \in \mathbb{N}$. Then we have $\tilde{\delta}^*(\tilde{q}_i,\phi(x\omega\sigma)) = \tilde{\delta}^*(\tilde{\delta}^*(\tilde{q}_i,\phi(x\omega)),\phi(\sigma)) = \tilde{\delta}^*(\tilde{\delta}^*(\tilde{q}_i,\phi(y\omega)),\phi(\sigma)) = \tilde{\delta}^*(\tilde{q}_i,\phi(y\omega\sigma))$. This completes the proof. □

Proposition 2: *For a marked realisation $\langle G_i, p_i\rangle$ of a DFSA G_i with a specified unobservability map p_i,*

1. *$\mathcal{N}_{\mathrm{p}}$ is a refinement of $\mathcal{N}$, i.e., $\mathcal{N}_{\mathrm{p}} \leq \mathcal{N}$.*
2. *If $\langle G_i, p_i\rangle$ is a marked realisation of the plant automaton G_i with an unobservability map p_i, then $\mathcal{R}_{\langle G_i,p_i\rangle} \leq \mathcal{N}_p$.*
3. *$\mathcal{N}_{\mathrm{p}}$ induces a marked realisation $\langle G_i, p_i\rangle_*$ with $\mathcal{R}_{\langle G_i,p_i\rangle_*} = \mathcal{N}_p$.*

4. *$\langle G_i, p_i\rangle_*$ is the unique (up to a renaming of states) minimal marked realisation in the sense $\mathcal{R}_{\langle G_i, p_i\rangle} \leq \mathcal{R}_{\langle G_i, p_i\rangle_\star}$.*

Proof:

1. It follows from Definitions 12 and 13 that $\mathcal{N}_{\text{p}}$ is a refinement of $\mathcal{N}$.
2. We need to show

$$\forall x, y \in \Sigma^*(x\mathcal{R}_{\langle G_i, p_i\rangle} y \Rightarrow x\mathcal{N}_p y)$$
$$\text{i.e. } \forall x, y \in \Sigma^*\left(\left(\tilde{\delta}^*(\tilde{q}_i, \phi(x)) = \tilde{\delta}^*(\tilde{q}_i, \phi(y))\right) \Rightarrow x\mathcal{N}_p y\right)$$

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

$$\delta(q_i, x) = \delta(q_i, y) = \text{MERGE}(\tilde{q}_k) \in Q \Longrightarrow x\mathcal{N}y. \tag{23}$$

Now, we claim

$$\forall u \in \Sigma^*(xu \in \overline{L_m(G_i)} \Rightarrow \mathfrak{p}_i^x(u) = \mathfrak{p}_i^y(u)). \tag{24}$$

For $|u| = 0$, the result is immediate from noting $\mathfrak{p}_i^x(\epsilon) = \mathfrak{p}_i^y(\epsilon) = \epsilon$. Hence we consider $|u| \geq 1$. We use induction on the length of u. Let $|u| = 1$. Assume $xu \in \overline{L_m(G_i)}$ and, if possible, let $\mathfrak{p}_i^x(u) = u \bigwedge \mathfrak{p}_i^y(u) = \epsilon$. Since $\delta^*(q_i, x) = \delta^*(q_i, y) = \text{MERGE}(\tilde{q}_k)$ and $xu \in \overline{L_m(G_i)}$, there exists $q_\ell \in Q$ such that $\delta(\text{MERGE}(\tilde{q}_k), u) = q_\ell$. Hence, it follows that $(\tilde{q}_k, (u, 0), q_{\ell 1}^{\mathcal{I}}) \in \tilde{\delta} \bigwedge (\tilde{q}_k, (u, 1), q_{\ell 2}^{\mathcal{I}}) \in \tilde{\delta}$ where $\text{MERGE}(q_{\ell 2}^{\mathcal{I}}) = \text{MERGE}(q_{\ell 1}^{\mathcal{I}}) = q_\ell$. 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 $xu\sigma \in \overline{L_m(G_i)}$. Then we have

$$\mathfrak{p}_i^x(u\sigma) = \mathfrak{p}_i^x(u)\mathfrak{p}_i^{xu}(\sigma) = \mathfrak{p}_i^y(u)\mathfrak{p}_i^{xu}(\sigma). \tag{25}$$

Since $\mathcal{R}_{\langle G_i, p_i\rangle}$ is right invariant (see Proposition 1), we have $\tilde{\delta}^*(\tilde{q}_i, \phi(xu)) = \tilde{\delta}^*(\tilde{q}_i, \phi(yu))$. Hence it follows from the same argument as given for the case $|u| = 1$, that $\mathfrak{p}_i^{xu}(\sigma) = \mathfrak{p}_i^{yu}(\sigma)$. Hence, we have

$$\mathfrak{p}_i^x(u\sigma) = \mathfrak{p}_i^y(u)\mathfrak{p}_i^{xu}(\sigma) = \mathfrak{p}_i^y(u)\mathfrak{p}_i^{yu}(\sigma) = \mathfrak{p}_i^y(u\sigma). \tag{26}$$

This completes the induction. Equations (23) and (26) implies that $x\mathcal{N}_{\text{p}}$y. For Case (2), Lemma 4 implies $x, y \notin \overline{L_m(G_i)}$. Prefix closure of $\overline{L_m(G_i)}$ implies that $\forall u \in \Sigma^*(xu, yu \notin \overline{L_m(G_i)})$ which in turn implies $x\mathcal{N}$y. Since the condition $\forall u \in \Sigma^*(xu \in \overline{L_m(G_i)} \Rightarrow \mathfrak{p}_i^x(u) = \mathfrak{p}_i^y(u))$ is vacuously satisfied (since there is no such u), we have $x\mathcal{N}_{py}$. This completes the proof.

3. Let $\mathscr{E}^p = \{E_k : k \in \mathcal{J} \subseteq \mathbb{N}\}$ be the set of equivalence classes of $\mathcal{N}_p$ where $\mathcal{J}$ is an appropriate index set. We construct a labelled deterministic transition system $\langle G_i, p_i\rangle_\star = (\mathscr{E}^p, \Sigma \times \{0, 1\}, \tilde{\delta}, \tilde{q}_i)$ as follows:

- $\tilde{q}_i = E_i$ where $\epsilon \in \text{E}_\text{i} \in \mathscr{E}^\text{p}$.
- $\exists x \in \Sigma^*((x \in E_k) \bigwedge (x\sigma \in E_j) \bigwedge (\mathfrak{p}_i^x(\sigma) = \epsilon)) \Rightarrow (E_k, (\sigma, 0), E_j) \in \tilde{\delta}$
- $\exists x \in \Sigma^*((x \in E_k) \bigwedge (x\sigma \in E_j) \bigwedge (\mathfrak{p}_i^x(\sigma) = \sigma)) \Rightarrow (E_k, (\sigma, 1), E_j) \in \tilde{\delta}$
- There are no other elements in $\tilde{\delta}$.

Next we show that $\langle G_i, p_i\rangle_*$ is a marked realisation in the sense of Definition 11. Since $\mathcal{N}_{\text{p}} \leq \mathcal{N}$, there exists a surjective map $\Psi: \mathscr{E}^p \to \text{Q}$. This serves as the merging map for $\langle G_i, p_i\rangle_*$. Next we claim $\exists x \in E_k((x\sigma \in E_j) \bigwedge (\mathfrak{p}_i^x(\sigma) = \epsilon)) \Rightarrow \quad \forall x \in E_k((x\sigma \in E_j) \bigwedge (\mathfrak{p}_i^x(\sigma) = \epsilon))$. 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 $\mathcal{N}_p$ (See Lemma 2) that $x\sigma\mathcal{N}_p y\sigma \to y\sigma \in E_j$. Also, $x\mathcal{N}_p y \Rightarrow \mathfrak{p}_i^x(\sigma) = \mathfrak{p}_i^y(\sigma)$. Hence we conclude $(E_k, (\sigma, 0), E_j) \in \tilde{\delta} \Rightarrow (E_k, (\sigma, 1), E_r) \notin \tilde{\delta} \forall E_r \in \mathscr{E}^p$. Similarly, $(E_k, (\sigma, 1), E_j) \in \tilde{\delta} \Rightarrow (E_k, (\sigma, 0), E_r) \notin \tilde{\delta} \forall E_r \in \mathscr{E}^p$. Thus Properties 2 and 3 of Definition 11 are satisfied. Property 4 is satisfied by construction as follows. We note that for $h \in \overline{L_m(G_i)}, \sigma \in \Sigma$, $\Psi(E_k) = \delta(q_i, h) \bigwedge \Psi(E_r) = \delta(\delta(q_i, h), \sigma) \bigwedge \mathfrak{p}_i^h(\sigma) = \epsilon \Leftrightarrow h \in E_k, h\sigma \in E_r$ and hence we have $(E_k, (\sigma, 0), E_r) \in \tilde{\delta}$. Similarly, if $\mathfrak{p}_i^h(\sigma) = \sigma$ then $(E_k, (\sigma, 1), E_r) \in \tilde{\delta}$. Thus $\langle G_i, p_i\rangle_*$ is indeed a marked realisation for plant G_i with the unobservability p_i. Finally, we show $\mathcal{R}_{\langle G_i, p_i\rangle *} = \mathcal{N}_{\text{p}}$. Since from item (2) above we have $\mathcal{R}_{\langle G_i, p_i\rangle *} \leq \mathcal{N}_{\text{p}}$, we only need to show the converse, i.e., to show $\mathcal{N}_{\text{p}} \leq \mathcal{R}_{\langle G_i, p_i\rangle *}$ for which we need to prove the following statement:

$$x\mathcal{N}_p y \Rightarrow \left(\tilde{\delta}^*(\tilde{q}_i, \phi(x)) = \tilde{\delta}^*(\tilde{q}_i, \phi(y))\right). \tag{27}$$

First, we claim that

$$x \in E_k \Rightarrow \tilde{\delta}^*(\tilde{q}_i, \phi(x)) = E_k. \tag{28}$$

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

$$\tilde{\delta}^*(\tilde{q}_i, \phi(\epsilon)) = \tilde{\delta}^*(\tilde{q}_i, \epsilon) = \tilde{q}_i = E_i \tag{29}$$
$$\epsilon \in E_i \text{ (By Definition of } \mathscr{E}^p \text{ above).} \tag{30}$$

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

$$x \in E_k \Rightarrow \tilde{\delta}^*(\tilde{q}_i, \phi(x)) = E_k \tag{31}$$
$$\Rightarrow (x\sigma \in E_j \Rightarrow \forall y \in E_k(y\sigma \in E_j)) \tag{32}$$

$$\Rightarrow \tilde{\delta}^*(\tilde{q}_i, \phi(x\sigma)) = \tilde{\delta}^*(\tilde{\delta}^*(\tilde{q}_i, \phi(x)), \phi(\sigma))$$
$$= \tilde{\delta}^*(E_k, \phi(\sigma)) = E_j.$$

Now, let $x, y \in \Sigma^*$ such that $x\mathcal{N}_{py}$, i.e., $(x\mathcal{N}y \bigwedge(\forall u \in \Sigma^*(xu \in \overline{L} \Rightarrow (\mathfrak{p}_i^x(u) = \mathfrak{p}_i^y(u)))))$. It follows that $x, y \in E_k \in \mathscr{E}^p$ which in turn implies from Equation (28) that $\tilde{\delta}^*(\tilde{q}_i, \phi(x)) = \tilde{\delta}^*(\tilde{q}_i, \phi(y))$ and hence we have $x\mathcal{R}_{\langle G_i,p_i\rangle *}y$. 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 $\mathcal{R}_{\langle G_i,pi\rangle} \leq \mathcal{R}_{\langle G_i,pi\rangle *}$. Assume there exists a marked realisation $\langle G_i, p_i\rangle_1$ such that $\mathcal{R}_{\langle G_i,p_i\rangle 1} = \mathcal{R}_{\langle G_i,p_i\rangle *}$. From item (3) it follows $\mathcal{R}_{\langle G_i,p_i\rangle 1} = \mathcal{N}_p$. Hence there exists a bijective mapping between the equivalence classes of $\mathcal{R}_{\langle G_i,p_i\rangle 1}$ and that of $\mathcal{N}_p$. Hence, $\langle G_i, p_i\rangle_*$ is the minimal realisation unique up to renaming of states. □

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 Σ 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*, $\mathcal{N}_p = \mathcal{N}$.

Proof: Let E_k be an equivalence class of the Nerode equivalence relation $\mathcal{N}$. State-dependent unobservability implies that if $\sigma \in \Sigma$ is defined from the state corresponding to the equivalence class E_k, $\forall x, y \in E_k$, $\mathfrak{p}_i^x(\sigma) \equiv \mathfrak{p}_i^y(\sigma)$. Let the event σ lead the automaton to a state that corresponds to the equivalence class E_j. Then, $\forall x, y \in E_k$, $\mathfrak{p}_i^x(u) \equiv \mathfrak{p}_i^y(u)\ \forall u \in \Sigma^*$ such that $xu \in L$ and hence, $x\mathcal{N}y \Rightarrow x\mathcal{N}_p y \forall x, y \in L$, $\Rightarrow \mathcal{N} \leq \mathcal{N}_p$

By Proposition 2, we have $\mathcal{N}_p \leq \mathcal{N} \Rightarrow \mathcal{N} \equiv \mathcal{N}_p$. □

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 $\mathcal{N}_p$ is a refinement of the Nerode equivalence $\mathcal{N}$. An equivalence class under $\mathcal{N}$ is possibly split further under $\mathcal{N}_p$. If the jth equivalence class of $\mathcal{N}$ is denoted by $\mathcal{N}^j$, and the total number of splits that $\mathcal{N}^j$ undergoes under $\mathcal{N}_p$ is denoted by $[\mathcal{N}^j : p]$, then we have:

Lemma 5: $\langle G_i, p_i\rangle_\star \uparrow \Leftrightarrow (\exists j \in \mathcal{I}_Q,\ s.t.\ [\mathcal{N}^j : p]\ \uparrow)$

Proof: $\langle G_i, p_i\rangle_\star \uparrow \iff \exists q_j \in Q(\text{MERGE}^{-1}(\{q_j\}) \uparrow) \iff \exists q_j \in Q([\mathcal{N}^j : p] \uparrow)$ □

Proposition 4: *Let* **REG** *denote the set of regular languages for an DFSA G_i and let p_i be an unobservability map and $\mathbb{O}_p$ be the corresponding observed language. If there exists a finite minimal realisation* $\langle G_i, p_i\rangle_*\downarrow$ induced by $\mathcal{N}_p$, *then $\mathbb{O}_p$ is regular. That is* $\langle G_i, p_i\rangle_\star \downarrow \Rightarrow \mathbb{O}_p \in$ **REG**.

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 completion. Since $\langle G_i, p_i\rangle_*$ is finite, the marked transitions can be replaced by ϵ-transitions. The resultant automaton is a finite state machine with ϵ-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 $\mathbb{O}_p$ 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 $\mathbb{P}_\infty$ if the following statement is true.

$$\forall h \in \overline{L_m(G_i)},\ \forall \sigma \in \Sigma,\ \forall u \in \Sigma^*$$
$$\Big(((h\sigma u \in \overline{L_m(G_i)}) \bigwedge (\mathfrak{p}_i^h(\sigma) = \epsilon)\Big) \Longrightarrow \big(\mathfrak{p}_i^h(\sigma u) = \epsilon\big)\Big)$$

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

Proposition 5: $((\langle G_i, p_i\rangle_\star \uparrow \bigwedge \mathbb{P}_\infty) \Rightarrow (\mathbb{O}_p \notin \mathbf{REG})$

Proof: Since the p-minimal realisation $\langle G_i, p_i\rangle_*\uparrow$ is an infinite realisation, it follows from Lemma 5, that $\exists q_k \in Q([\mathcal{N}^k : p]\uparrow)$, i.e., there exists a $q_k \in G$, which is split into (countably) infinite number of states $\{q_{k_1}, q_{k_2}, \ldots, q_{k_j}, \ldots\}$ 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

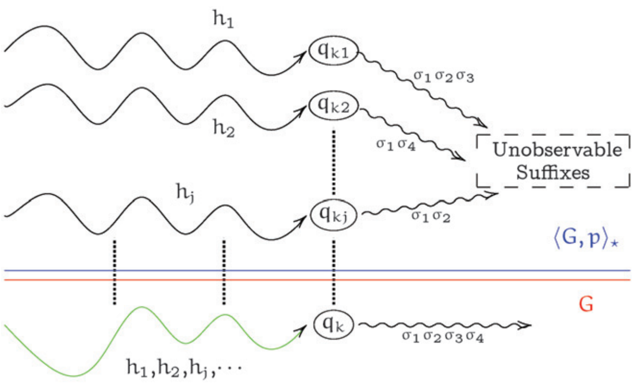

Figure 5. The minimal realisation is above while the underlying plant is below the double lines. State q_k in the underlying plant is split into infinitely many $q_{k1}, q_{k2}, \ldots$ in the realisation. Strings $h_1, h_2, \ldots$ terminate on the same state q_k in the underlying plant. In the minimal realisation each h_j terminates on a distinct q_{kj}.

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 $\{q_{k_1}, q_{k_2}, \ldots, q_{k_j}, \ldots\}$ must collapse to a finite set and the condition $\langle G_i, p_i\rangle_* \uparrow$ will be violated. The fact that one may choose $h_1, h_2, \ldots, h_j, \ldots$ follows from the axiom of choice. Now, $\mathbb{P}_\infty$ implies that at most one string from the set $\{h_1, h_2, \ldots, h_j, \ldots\}$ 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_j} 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_{k_j} and q_{k_l} must collapse; this violates the condition that $\langle G_i, p_i\rangle_*$ 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_j} from $\{q_{k_1}, q_{k_2}, \ldots\}$. Let us denote the new string set $\{h_1, h_2, \ldots\}$ by $\mathcal{H}$ and the new state set $\{q_{k_1}, q_{k_2}, \ldots\}$ 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 $h_1 \mathcal{N} h_2$. As h_1 and h_2 lead to different states in the p-minimal realisation $\langle G_i, p_i\rangle_*$, it follows $h_1 \rvert\mathcal{N}_p h_2$. Now if L is the accepted language of the automaton, $(h_1 \mathcal{N}_p h_2) \Rightarrow$ $(\exists u \in \Sigma^*$ such that $h_1 u, h_2 u \in L)$ and $(\mathfrak{p}_i^{h_1}(u) \neq \mathfrak{p}_i^{h_2}(u))$.

Now, $\mathbb{P}_\infty \Rightarrow ((\mathfrak{p}_i^{h_1}(u) = \omega_1) \bigwedge (\mathfrak{p}_i^{h_2}(u) = \omega_2))$, where $u \equiv \omega_1 v_1 \equiv \omega_2 v_2$, $v_1, v_2 \in \Sigma^*$ with $\omega_1 \neq \omega_2$. This implies that there exists $\omega_3 \in \Sigma^*$ such that either $\omega_2 \equiv \omega_1\omega_3$ or $\omega_1 \equiv \omega_2\omega_3$. Assuming that $\omega_2 \equiv \omega_1\omega_3$, it follows that $((h_1\omega_1\omega_3 \notin \mathbb{O}_p) \bigwedge (h_2\omega_1\omega_3 \in \mathbb{O}_p))$. By the Nerode equivalence relation, it is concluded that h_1 and h_2 lead to distinct states in any realisation for $\mathbb{O}_\mathrm{p}$. Since h_1 and h_2 are two arbitrary stings in the set $\mathcal{H}$, each element of $\mathcal{H}$ must lead to a distinct state in any realisation for $\mathbb{O}_\mathrm{p}$. Infinite cardinality of $\mathcal{H}$ implies $\mathbb{O}_\mathrm{p} \notin$ **REG**. □

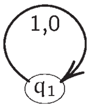

Figure 6. The plant automaton.

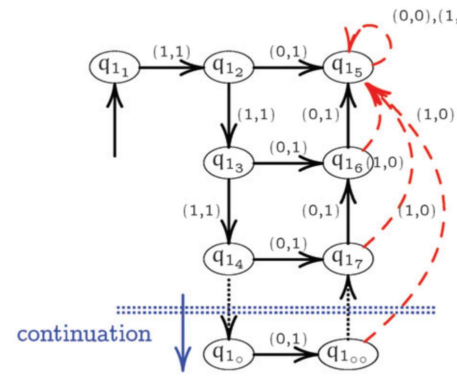

Figure 7. p-minimal realisation with observable transitions in solid lines.

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 $\mathbb{O}_p$ is $\{1^n 0^m : m \leq n$ and $m, n \in \mathbb{N}\}$. Hence, $\mathbb{O}_\mathrm{p} \notin$ **REG** is in accordance with Proposition 5.

Next it is shown that an infinite p-minimal realisation does not guarantee that the observed language $\mathbb{O}_\mathrm{p}$ is non-regular.

Definition 18: 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 $\mathbb{P}_{\mathcal{L}oop}$ *if* p_i satisfies the following criterion:

$$\forall h \in \overline{L_m(G_i)}, \forall \sigma \in \Sigma,\ \left(\mathfrak{p}_i^h(\sigma) = \epsilon\right)$$
$$\Longrightarrow (\exists v_1 \in \Sigma^*, \tau \in \Sigma,\ \text{s.t.}(\mathfrak{p}_i^{h\sigma}(v_1) = \epsilon)$$
$$\bigwedge (\mathfrak{p}_i^{h\sigma v_1}(\tau) = \tau) \bigwedge (h\ \mathcal{N}\ h\sigma v_1)).$$

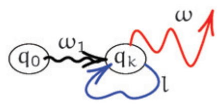

Figure 8. Situation under $\mathbb{P}_{\mathcal{L}oop}$.

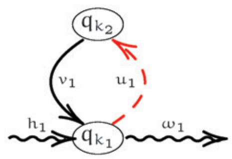

Figure 9. Scenario #1.

Remark 4: The condition $\mathbb{P}_{\mathcal{L}oop}$ 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_i, satisfies the condition $\omega \mathcal{N} \omega l$.) This condition follows from the requirement of $\mathbb{P}_{\mathcal{L}oop}$ 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 $\mathbb{P}_{\mathcal{L}oop}$ is violated (i.e., there may not exist a τ 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,* $\mathbb{P}_{\mathcal{L}oop} \Longleftrightarrow (\mathbb{O}_p = \overline{L_m(G_i)})$.

Proof: To prove $\mathbb{P}_{\mathcal{L}oop} \Longrightarrow (\mathbb{O}_p = \overline{L_m(G_i)})$, we proceed as follows. Let $\overline{L_m(G_i)} = L$. It is given that $\mathbb{P}_{\mathcal{L}oop} \Rightarrow (\forall u \in \Sigma^*, p_i(u) \equiv p_i(\omega_1 l^m \omega) = \omega_1 l^n \omega \in \mathbb{O}_p,$ with $m \geq n \in \mathbb{N})$, where $\omega_1, \omega, l \in \Sigma^*$, and l is a loop. Since l is a loop, $(\omega_1 l^m \omega \in L) \Rightarrow (\omega_1 l^k \omega \in L \forall k \leq m) \Rightarrow (\omega_1 l^n \omega \in L)$. Hence, $(p(u) \in \mathbb{O}_p) \Rightarrow (p(u) \in L) \Rightarrow (\mathbb{O}_p \subseteqq L)$. Also, $((\omega_1 l^m \omega \in L) \bigwedge (\omega_1 l^m \omega \longmapsto \omega_1 l^n \omega) \bigwedge \mathbb{P}_{\mathcal{L}oop}) \Rightarrow (\exists k \in \mathbb{N} | \omega_1 l^m l^k \omega \longmapsto \omega_1 l^{n+1} \omega)$. Hence, by induction, $\exists k_\circ \in \mathbb{N}$ such that $(\omega_1 l^m l^{k_\circ} \omega \longmapsto \omega_1 l^m \omega) \Rightarrow (u \in L) \Rightarrow (u \in \mathbb{O}_p)$, i.e., $L \subseteqq \mathbb{O}_p$. Therefore $L \equiv \mathbb{O}_\mathrm{p}$.

To prove the converse, let $\mathbb{O}_P \equiv L$ and let us assume $\neg \mathbb{P}_{\mathcal{L}oop}$. 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 $p(h_1 u_1 v_1) = h_1 v_1 \in \mathbb{O}_\mathrm{P}$. Now, $\mathbb{O}_P \equiv L$ 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 ω_2 initiating from state q_{k_2}. Hence, it follows that $\forall \omega \in \Sigma^*$ $(h_1 u_1 \omega \in L) \Leftrightarrow (h_1 \omega \in L)$. Hence, q_{k_1} and q_{k_2} collapse to the same state in the

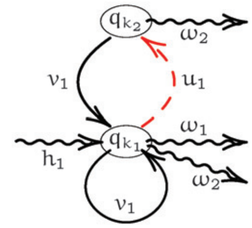

Figure 10. Scenario #2.

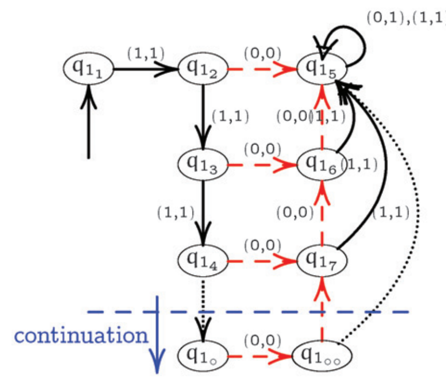

Figure 11. p-minimal realisation with observable transitions in solid lines.

minimal realisation of the language L, which implies that u_1 is a loop. The entire loop u_1 cannot be unobservable because then $\mathbb{O}_P \neq L$. Hence, $(\mathbb{O}_P \equiv L) \Longrightarrow \mathbb{P}_{\mathcal{L}oop}$. □

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 $\mathbb{P}_{\mathcal{L}oop}$. 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 $\mathbb{O}_p \equiv \Sigma^*$. Hence $\mathbb{O}_p \equiv L$, 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 $((\langle G_i, p_i \rangle_\star) \uparrow \bigwedge \mathbb{P}_{\mathcal{L}oop}) \Rightarrow (\mathbb{O}_P \equiv L)$.

4. Σ-normal representation

This section investigates a uniform representation for unobservability maps. As before, the plant is assumed

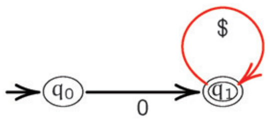

Figure 12. Augmenting trim plant automaton by addition of self-loop.

to be trim and the plant language regular. Specifically, let $G_i \equiv \langle Q, \Sigma, \delta, q_i, Q\rangle$ be a trim (i.e., accessible and co-accessible) finite-state automaton model that models the discrete-event dynamics of a physical plant, where $Q=\{q_k : k \in \mathcal{I}_Q\}$ is the set of states. In addition to trimness, we assume that the plant model satisfies the following property

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

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

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

where \$ is a symbol not in Σ 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 ($\mathscr{H}_\sigma$) is defined as:

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

Definition 20: For an event $\sigma \in \Sigma$, and a given state $q_k \in Q$, the relative history at state q_k, ($\mathscr{H}_\sigma|_{q_k}$) is defined as.: $\mathscr{H}_\sigma|_{q_k} \equiv \{\omega \in \overline{L_m(G_i)} \ : \ \delta^*(q_i,\omega) = q_k \bigwedge \delta(\delta^*(q_i,\omega),\sigma) \in Q\}$.

The following points are worth noting.

- We have

$$\mathscr{H}_\sigma = \bigcup_{q_k \in Q} \mathscr{H}_\sigma|_{q_k} \quad (35a)$$

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

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 $\sigma_1, \sigma_2 \in \Sigma$, $\mathscr{H}_{\sigma_1}|_{q_k}$ and $\mathscr{H}_{\sigma_2}|_{q_k}$ are not necessarily disjoint. In fact, it follows that if both σ_1 and σ_2 are defined from the state q_k, we have $\mathscr{H}_{\sigma_1}|_{q_k} = \mathscr{H}_{\sigma_2}|_{q_k} \equiv L(q_i, q_k)$ and if there is no σ_1 defined from q_k, then $\mathscr{H}_{\sigma_1}|_{q_k} = \oslash$.
- Thus, $\mathscr{H}_{\sigma_1}|_{q_k} \in \{\oslash, L(q_i, q_k)\}$.

Definition 21: For a given alphabet $\Sigma = \{\sigma_0, \ldots, \sigma_{m-1}\}$ and an unobservability map p, the Σ-normal phantom set $\mathbb{L}$ is a set of languages $\{\mathcal{L}_{\sigma_0}, \ldots, \mathcal{L}_{\sigma_i}, \ldots, \mathcal{L}_{\sigma_{m-1}}\}$ such that, $(\forall \sigma_i \in \Sigma(\mathcal{L}_{\sigma_i} \subseteq \mathscr{H}_{\sigma_i})) \bigwedge ((\forall \omega \in \mathscr{H}_{\sigma_i}(\mathfrak{p}_i^\omega(\sigma_i) = \epsilon)) \Leftrightarrow (\omega \in \mathcal{L}_{\sigma_i}))$

Proposition 7 (Σ-normal Representation Theorem)**:** *The Σ-normal phantom set $\mathbb{L}$ is uniquely specified by the unobservability map p. Conversely, any set of languages $\{L_{\sigma_0}, \ldots, L_{\sigma_{m-1}}\}$ which satisfies $L_{\sigma_i} \subseteq \mathscr{H}_{\sigma_i}\ \forall \sigma_i \in \Sigma$ uniquely corresponds to an unobservability map p.*

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

$$\left(\overline{L_m(G_i)} = \bigcup_{\sigma\in\Sigma} \mathscr{H}_\sigma\right) \Longrightarrow \left(\overline{L_m(G_i)} = \bigcup_{\sigma\in\Sigma}\left(L_{\sigma_i} \cup (\mathscr{H}_{\sigma_i} \setminus L_{\sigma_i})\right)\right). \quad (36)$$

Equation (36) allows us to define a map $\wp : \Sigma^* \times \Sigma \longrightarrow \Sigma \bigcup \{\epsilon\}$ as follows:

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

The map $\wp$ is well-defined because $(\mathscr{H}_{\sigma_i} \setminus L_{\sigma_i}) \bigcap L_{\sigma_i} = \oslash$. Next we define an induced map $\widetilde{\wp} : \Sigma^* \longrightarrow \Sigma^*$ as follows:

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

$\widetilde{\wp}$ satisfies the basic properties of an unobservability map, namely, $\widetilde{\wp}(\sigma) \in \{\epsilon, \sigma\}$ by definition of $\wp$ and $\widetilde{\wp}(\omega\sigma) = \widetilde{\wp}(\omega)\wp(\omega,\sigma) \in \left\{\widetilde{\wp}(\omega)\sigma, \widetilde{\wp}(\omega)\right\}$. Hence, there exists at least one unobservability map $p \equiv \widetilde{\wp}$ which corresponds to $\mathscr{L}$. Let, if possible, p^1 and p^2 be two such maps with $p^1 \neq p^2$. Now, $(p^1 \neq p^2) \Longrightarrow (\exists\ \omega \in \Sigma^*, \sigma \in \Sigma,$ such that $p^1_\omega(\sigma) \neq p^2_\omega(\sigma))$. Without loss of generality, it can be assumed that $p^1_\omega(\sigma) = \epsilon$ and $p^1_\omega(\sigma) = \sigma$. Now, $(p^1_\omega(\sigma) = \epsilon) \Longrightarrow (\omega \in L_\sigma)$ and $(p^2_\omega(\sigma) = \sigma) \Longrightarrow (\omega \in (\mathscr{H}_\sigma \setminus L_\sigma))$. But $\forall \sigma_i \in \Sigma((\mathscr{H}_{\sigma_i} \setminus L_{\sigma_i}) \bigcap L_{\sigma_i} = \oslash)$

which implies $\mathfrak{p}^1$ and $\mathfrak{p}^2$ coincide everywhere (contradiction). Hence, $\mathfrak{p}$ is unique. □

Remark 6: It follows from Proposition 7 that the Σ-normal phantom set $\mathbb{L}$ (See Definition 21) is in fact a 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 $\overline{L_m(G_1)} = \{0, 1\}^*$ 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 Σ-normal phantom set in this case is as follows:

$$\mathbb{L} = \{\mathcal{L}_0, \mathcal{L}_1\} \text{ where} \begin{cases} \mathcal{L}_0 & = 1^*(01^*01^*)^* \\ \mathcal{L}_1 & = \oslash. \end{cases}$$

$\mathcal{L}_1$ is empty since 1 is always observable. $\mathcal{L}_0$ 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 $1^* \subset 1^*(01^*01^*)^* \equiv \mathcal{L}_0$. Note that the p-minimal realisation is finite and every element of $\mathbb{L}$ 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 Σ-normal phantom set consists of ℓ copies of the null set, where ℓ is the cardinality of the alphabet, i.e., $p = \mathfrak{id} \Longrightarrow \mathbb{L} = \{\oslash, \ldots, \oslash\}$. This follows from the fact that if an event σ is always observable, irrespective of its past, then $\mathcal{L}_\sigma = \ominus$.
- For a completely unobservable system, $\mathbb{L} = \{\mathscr{H}_\sigma : \sigma \in \Sigma\}$. This follows from the fact that if an event σ is always unobservable, then the set of histories for which σ is unobservable is, in fact, the entire set of histories for the event i.e. $\mathscr{H}_\sigma$ by definition.

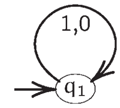

Figure 13. Trim plant model.

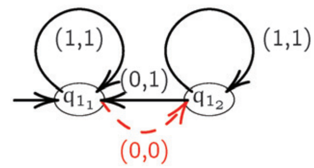

Figure 14. p-minimal realisation: Unobservable transition in dashed line.

- State independent unobservability can be identified with the following case: $\mathbb{L} = \{\mathcal{L}_\sigma : \mathcal{L}_\sigma \in \{\mathscr{H}_\sigma, \oslash\}\}$. It is obvious that such a phantom set implies that an event is either always observable or always unobservable which is precisely the requirement of state independent unobservability. Note that all elements of the phantom set are regular.
- State dependent unobservability can be identified with the following case: $\mathbb{L} = \{\mathcal{L}_\sigma : \mathcal{L}_\sigma = \bigcup_{k \in \mathcal{B}_\sigma \subseteq \mathcal{I}_Q} \mathscr{H}_\sigma|_{q_k}\}$ where $\mathcal{B}_\sigma$ is some arbitrary subset of $\mathcal{I}_Q$. Note that $\mathcal{B}_\sigma$ can be empty and hence $\mathcal{L}_\sigma$ can be empty as well. Also all elements of the phantom set are still regular.

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

Proposition 8: $\forall \sigma \in \Sigma, (\mathcal{L}_\sigma \in \mathbf{REG}) \Longleftrightarrow \langle G_i, p_i \rangle_\star \downarrow$.

Proof: First we show, $\langle G_i, p_i \rangle_* \uparrow \Rightarrow \exists \sigma \in \Sigma$, such that $\mathcal{L}_\sigma \notin \mathbf{REG}$. Assume $\langle G_i, p_i \rangle_* \uparrow$. This implies that the projective Nerode equivalence relation is not of finite index. Hence it follows that there exists $\mathcal{H} = \{h_1, h_2, \ldots, h_r, \ldots\}$ with $r \in \mathbb{N}$ such that $\forall h_r, h_j \in \mathcal{H}$, $\exists u \in \Sigma^*$ such that $\mathfrak{p}_i^{h_r}(u) \neq \mathfrak{p}_i^{h_j}(u)$. Now, finiteness of Σ implies that for some $\sigma \in \Sigma$, there exists an infinite set $\mathcal{H}_\sigma \subseteq \mathcal{H}$ such that $\forall h_r, h_j \in \mathcal{H}_\sigma, \exists u \in \Sigma^*$ with $\mathfrak{p}_i^{h_r u}(\sigma) \neq \mathfrak{p}_i^{h_j u}(\sigma)$. Denote the Nerode equivalence on Σ^* induced by $\mathcal{L}_\sigma$ by $\mathcal{N}_{\mathcal{L}_\sigma}$. We note that $x \mathcal{N}_{\mathcal{L}_\sigma} y \Rightarrow \forall u \in \Sigma^*(xu \in \mathcal{L}_\sigma \Leftrightarrow yu \in \mathcal{L}_\sigma)$. Let $\tilde{h}_r, \tilde{h}_j \in \mathcal{H}_\sigma$. Then we have $\tilde{h}_r \mathcal{N}_{\mathcal{L}_\sigma} \tilde{h}_j$, but for some $u \in \Sigma^*$, $\tilde{h}_r u \; \mathcal{N}|_{\mathcal{L}_\sigma} \tilde{h}_j u$. Infinite cardinality of $\mathcal{H}_\sigma$ implies $\mathcal{N}_{\mathcal{L}_\sigma}$ is of infinite index. Hence $\mathcal{L}_\sigma \notin \mathbf{REG}$. The converse follows immediately by noting that if $\langle G_i, p_i \rangle_* \downarrow$, then a finite state description for $\mathcal{L}_\sigma$ is obtained by defining as accepting states all states (of $\langle G_i, p_i \rangle_*$) at which the event $(\sigma, 0)$ is defined and replacing the labels (σ, b) with σ. (Note $b \in \{0, 1\}$). This completes the proof. □

Table I. Hierarchy of unobservability maps.

All elements of $\mathbb{L}$ are regular	Regular unobservability
All elements of $\mathbb{L}$ are context free	Context free unobservability
All elements of $\mathbb{L}$ are context sensitive	Context sensitive unobservability
All elements of $\mathbb{L}$ are recursively enumerable	Recursively enumerable unobservability

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 $G \equiv \langle Q, \Sigma, \delta, q_i, Q_m \rangle$ with $\Sigma = \{\sigma_0, \ldots, \sigma_{m-1}\}$ and a specified Σ-normal phantom set $\mathbb{L} = \{\mathcal{L}_0, \ldots, \mathcal{L}_{m-1}\}$. 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 $\overline{Q} : p(\overline{L(G_i)}) \longrightarrow 2^Q$ 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 ω, $\overline{Q}(\omega) \subseteq Q$ 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 ω and a fixed state $q_j \in Q$, the state determinacy problem is the determination of the truth value of the following logical sentence:

$$\mathbb{D}(q_j, \omega)(q_j \in \overline{Q}(\omega)). \tag{37}$$

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

$$\mathbb{D}(q_j, \epsilon) := \mathbb{D}(q_j) := (q_j \in \overline{Q}(\epsilon)). \tag{38}$$

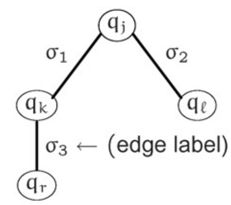

Figure 15. Tree: q_j is an ancestor of q_r; q_ℓ is not.

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 $\mathcal{T}(q_j)$

Input: $G, \mathbb{L}, q_j$
Output: $\mathscr{T}(q_j)$

```
begin
  Define the root node and label the root as q_j; /* Initiate Tree Construction */
  For each new node q_k in the frontier of the current tree do
  Compute A = CHILD(q_k); /* Compute possible child nodes */
  For each (q_ℓ, σ) ∈ A
    If q_k has an ancestor q_ℓ then
      Delete (q_ℓ, σ) from A /* Avoid repetition in the same ancestry (See Figure 15) /*
    endif
  endfor
  For each (q_ℓ, σ) ∈ A do
    Compute the path ω ∈ Σ* from the root to q_ℓ;
    Compute DEL_NODE(q_ℓ, ω);
    If DEL_NODE(q_ℓ, ω) = ⊖ then
      Delete (q_(ℓ,σ)) from A; /* Transitivity check
    end if
    For each remaining (q_r, σ) ∈ A do
      Create a node labelled q_r connected by a edge labelled Σ to q_k;
    endfor
  endfor
  Terminate if no new child nodes can be created;
  endfor
end
```

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

- $\textsc{Child} : Q \rightarrow 2^{Q \times \Sigma}$ such that $\forall q_k, q_j \in Q, \sigma \in \Sigma$,

$$(q_k, \sigma) \in \textsc{Child}(q_j) \iff \exists \sigma \in \Sigma | \delta(q_k, \sigma) = q_j \bigwedge q_k \neq q_j. \tag{39}$$

- $\text{DEL\_NODE} : Q \times \Sigma^* \to 2^{\Sigma^*}$ such that $\forall \sigma_i \in \Sigma, q_k \in Q$,

$$\begin{aligned}\text{DEL\_NODE}(q_k, \sigma_1 \ldots \sigma_m) &= \left(\mathscr{H}_{\sigma_m}|_{q_k} \sigma_m \sigma_{m-1} \ldots \sigma_2\right) \\ &\quad \cap \mathcal{L}_{\sigma_1} \cap \left(\bigcap_{i=2}^{i=m} \mathcal{L}_{\sigma_i} \sigma_i \sigma_{i-1} \ldots \sigma_2\right). \end{aligned} \tag{40}$$

We note that $\forall q_k \in Q, \sigma \in \Sigma \text{DEL\_NODE}(q_k, \sigma) = \mathscr{H}_\sigma|_{q_k} \cap \mathcal{L}_\sigma$.
Algorithm 1 constructs a phantom reachability tree $\mathscr{T}(q_j)$ for a given state $q_j \in Q$.

Proposition 9:

1. $\forall q_j \in Q$, $\mathscr{T}(q_j)$ is finite.
2. $q_0 \in \mathscr{T}(q_j) \Leftrightarrow q_j \in \overline{Q}(\epsilon)$. (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 frontier has length bounded by Card(Q) (since each node label occurs at most once in every path from the root). Also, each node has at most Card(Σ) immediate children; implying $\mathscr{T}(q_j)$ 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 $\mathscr{H}_{\sigma_1}|_{q_1}\sigma_1 \subseteq \mathscr{H}_{\sigma_2}|_{q_2}$. Using this fact, we note that if $\sigma_1 \ldots \sigma_m$ is a path in the tree from the root labelled q_j to a node labelled q_0, then we have by construction: $(\mathscr{H}_{\sigma_m}|_{q_0}\sigma_m\sigma_{m-1}\ldots\sigma_2) \cap \mathcal{L}_{\sigma_1} \cap (\bigcap_{i=2}^{i=m} \mathcal{L}_{\sigma_i}\sigma_i\sigma_{i-1}\ldots\sigma_2) \neq \oslash \Longleftrightarrow \exists q_1, \ldots, q_{m-2} \in Q$ with $(\mathcal{L}_{\sigma_m} \cap \mathscr{H}_{\sigma_m}|_{q_0})\sigma_m \cdots \sigma_2 \bigcap (\mathcal{L}_{\sigma_{m-1}} \cap \mathscr{H}_{\sigma_{m-1}}|_{q_1})\sigma_{m-1} \cdots \sigma_2 \bigcap \cdots\cdots \bigcap (\mathcal{L}_{\sigma_2} \cap \mathscr{H}_{\sigma_2}|_{q_{m-2}})\sigma_2 \bigcap \mathcal{L}_{\sigma_1} \cap \mathscr{H}_{\sigma_1}|_{q_j} \neq \oslash \Longleftrightarrow \exists\, h \in \overline{L(G_0)}$ such that $h \in L(q_0, q_0)$ and $q_0 \xrightarrow{\sigma_m} q_1 \xrightarrow{\sigma_{m-1}} \cdots \xrightarrow{\sigma_2} q_{m-2} \xrightarrow{\sigma_1} q_j$ with $p_h(\sigma_m) = \epsilon \bigwedge p_{h\sigma_m} \times (\sigma_{m-1}) p_{h\sigma_m}(\sigma_{m-1}) = \epsilon \bigwedge \cdots \bigwedge p_{h\sigma_m \cdots \sigma_2}(\sigma_1) = \epsilon$ Hence $\mathscr{T}(q_j)$ has a node labelled q_0 iff exists a completely unobservable path $\sigma_m \ldots \sigma_1$ from q_0 to q_j, i.e. $q_j \in \overline{Q}(\epsilon)$. □

Proposition 10: *Given a DFSA plant model and a Σ-normal phantom set $\mathbb{L}$,* **DE** *is decidable if $\mathbb{L} \subset \mathscr{L}$ such that $\forall L_1, L_2 \in \mathscr{L}, R \in REG$ $((L_1 \bigcap L_2 = \oslash$ is decidable$) \bigwedge (L_1 \bigcap R = \oslash$ is decidable$) \bigwedge (\forall \omega \in \Sigma^*, L_1\omega \in \mathscr{L}))$, i.e.* **DE** *is decidable if the Σ-normal phantom set belongs to a family of languages $\mathscr{L}$ which is closed under concatenation with singleton strings and for which the emptiness-checking problem for intersection of elements in $\mathscr{L}$ and for intersection of elements in $\mathscr{L}$ with regular languages can be effectively solved.*

Proof: The result follows immediately by noting that these three properties guarantee that $\text{DEL\_NODE}(q_j, \omega)$ can be effectively computed, and hence the phantom tree $\mathscr{T}(q_j)$ can be constructed algorithmically for any $q_j \in Q$. □

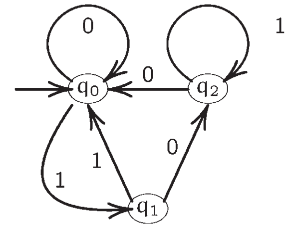

Figure 16. Plant model for Proposition 5.3.

Proposition 11: *Let* G_0 *be an arbitrary DFSA model and $\mathbb{L} \subseteq \mathscr{L}$ be the Σ-normal phantom set such that for the language-family $\mathscr{L}$, the emptiness-checking problem for intersection of two arbitrary languages is undecidable. Then* **DE** *is undecidable for this class of $\mathbb{L}$.*

Proof: Assume there exists an algorithm for $\mathscr{A}$ for solving **DE** for any arbitrary plant model such that the Σ-normal phantom set $\mathcal{L} \subset \mathscr{L}$. Further, we know that for two arbitrary languages $L_1, L_0 \in \mathscr{L}$, $L_1 \cap L_0 =_? \ominus$ is undecidable. 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 $L_1, L_0 \in \mathscr{L}$. We define $L_1{}' = L_1$ and $L_0{}' = L_0 1$. Now assume the Σ-normal phantom map satisfies the following criteria:

$$\begin{aligned}&(\mathcal{L}_1 \cap \mathscr{H}_1|_{q_1} = \phi) \bigwedge (\mathcal{L}_1 \cap \mathscr{H}_1|_{q_2} = \oslash) \\ &\bigwedge (\mathcal{L}_0 \cap \mathscr{H}_0|_{q_0} = \oslash) \bigwedge (\mathcal{L}_0 \cap \mathscr{H}_0|_{q_2} = \oslash) \\ &\bigwedge (\mathcal{L}_1 = L_1') \bigwedge (\mathcal{L}_0 = L_0'). \end{aligned} \tag{41}$$

Since, $\mathbb{L} \subset \mathscr{L}$, we use algorithm $\mathscr{A}$ to answer $q_2 \in_? \overline{Q}(\epsilon)$. We note that

$$\begin{aligned} q_2 \in_? \overline{Q}(\epsilon) &\equiv (\mathcal{L}_1 \cap \mathscr{H}_1|_{q_0})1 \bigcap (\mathcal{L}_0 \cap \mathscr{H}_0|_{q_1}) =_? \oslash \\ &\equiv (\mathcal{L}_1 1 \cap \mathcal{L}_0) \cap \mathscr{H}_1|_{q_0} 1 =_? \oslash. \end{aligned} \tag{42}$$

Since $(\mathcal{L}_1 \cap \mathscr{H}_1|_{q_1} = \oslash) \bigwedge (\mathcal{L}_1 \cap \mathscr{H}_1|_{q_2} = \oslash)$, it follows that $\mathcal{L}_1 \subseteq \mathscr{H}_1|_{q_0}$ implying

$$\begin{aligned} &(\mathcal{L}_1 1 \cap \mathcal{L}_0) \cap \mathscr{H}_1|_{q_0} 1 =_? \oslash \equiv (\mathcal{L}_1 1 \cap \mathcal{L}_0) =_? \oslash \\ &\quad \equiv (L_1' 1 \cap L_0') =_? \oslash \equiv (L_1 1 \cap L_0 1) =_? \oslash \equiv (L_1 \cap L_0) \\ &\quad =_? \phi \end{aligned}$$

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

Corollary 1: **DE** *and* **DG** *are undecidable for the class of problems where $\mathbb{L}$ is a set of arbitrary context-free languages.*

Proof: For two arbitrary context-free languages $L_1, L_2, L_1 \cap L_2 =_? \ominus$ 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 $q_j \in_? \overline{Q}(\omega)$ for all $\omega \in \Sigma^*$. Hence one can use the same algorithm to answer $q_j \in_? \overline{Q}(\epsilon)$ 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 $\mathbb{L} \subset \mathrm{REG}$. 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:l Petri net observer for reg. unobservability

Input: $\langle G, p\rangle$
Output: Petri net observer
begin
  I. Create a place q_j for each state q_j in $\langle G, p\rangle$;
  II. The set of transition labels is Σ;
  For *each* **observable** *transition* $q_j \xrightarrow[\sigma]{} q_k$ *in* $\langle G, p\rangle$ **do**
    I. Set the initial state in $\langle G, p\rangle$ to q_k;
    II. Compute $\overline{Q}(\epsilon)$;
    III. Add a transition labelled σ from the place q_j with output arcs to all places $q_l \in \overline{Q}(\epsilon)$;
  endfor
  For *each place* q_j *in the net* **do**
    For *each event* $\sigma \in \Sigma$ **do**
      If *there is no transition with label* σ *from* q_j **then**
        I. Add a flush-out arc with label σ from q_j
      endif
    endfor
  endfor
end

1. The Petri net formalism is natural, due to its ability to model transitions of the type $q1 \rightarrow \genfrac{}{}{0pt}{}{\nearrow q2}{\searrow q3}$, which reflects the condition "the plant can possibly be in states $q2$ or $q3$ 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 computed by executing 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: $\overline{Q}(\omega)$
begin
  I. Compute the initial marking for the observer as follows:
    a. Compute $\overline{Q}(\epsilon)$;
    b. Put a token in each place $q_j \in \overline{Q}(\epsilon)$;
  For j = 1 to r **do**
    I. Fire all enabled transitions labelled as τ_j in the observer;
    For *each place* q_j *in the observer* **do**
    if *number of tokens in* $q_j > 0$ **then**
      I. Normalise the number of tokens in q_j **to** 1.
    endif
    endfor
  endfor
  II. $\overline{Q}(\omega) = \{q_j \mid q_j$ has one token$\}$;
end

Proof: The complexity claim (Assertion 1) follows immediately from noting that the only non-trivial step is the computation of $\overline{Q}(\epsilon)$ 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 $\overline{Q}(\epsilon)$ (See Definition 22). If an event Σ is observed, Algorithm 2 implies that firing all σ labelled transitions in the

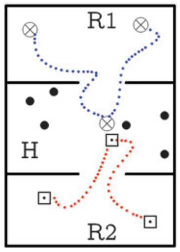

Figure 17. The game.

Table 2. State and event descriptions.

	STATES
00	Players at respective homes
01	INTRUDER in H, PATROL at home
11	Both players in H
10	INTRUDER at home, PATROL in H
	EVENTS
e	INTRUDER enters H
r	INTRUDER enters $R2$
a	PATROL switches room

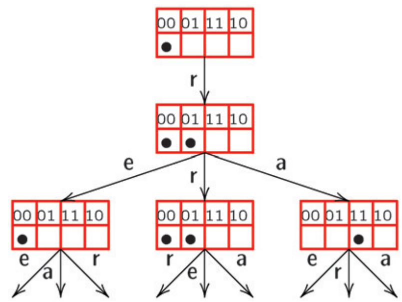

Figure 18. Sequential token distributions (first few possibilities) for the Petri net observer in situation 1.

observer is equivalent to computing the set of states that the plant may reach from each possible current state. Furthermore, if q_j is marked initially and σ is not defined from q_j in the underlying plant, then observing σ implies that the plant was not in q_k. This is taken care of by the σ-*labelledppp* flush-out arc in the observer (exists by construction), which flushes out or eliminates tokens from the place q_j. 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 ⊗) and INTRUDER (denoted by ⊡) 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 Σ-normal phantom set is given by $\mathbb{L} = \{\mathcal{L}_1 = \{\text{all strings terminating at state } 00\}, \oslash, \oslash\}$. (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.

$$\{1,1,0,0\} \underset{r}{\rightarrow} \{1,1,0,0\} \underset{a}{\rightarrow} \{0,0,1,0\} \underset{r}{\rightarrow} \{1,1,0,0\} \rightarrow \cdots \tag{43}$$

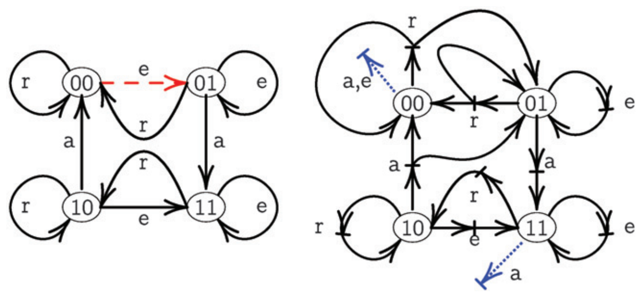

Figure 19. p-minimal realisation (left) and corresponding Petri net observer (right) in situation 1.

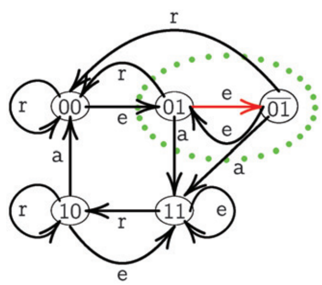

Figure 20. p-minimal realisation for situation 2.

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 $\overline{01}$. The phantom set in this case is given by $\mathbb{L} = \{\mathcal{L}_2, \oslash, \oslash\}$ where $\mathbb{L}_2 = \{$all strings terminating at state 01$\}$. Note that $\mathcal{L}_2$ is a regular subset of $\mathcal{L}_1$ 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.

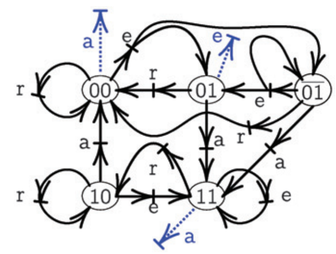

Figure 21. Constructed Petri net observer for situation 2.

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

Chattopadhyay, I., and Ray, A. (2007a), 'Language-measure-theoretic Optimal Control of Probabilistic Finite-state Systems,' *International Journal of Control*, 80, 1271–1290.

Chattopadhyay, I., and Ray, A., (2007b), 'Generalised Projections in Finite State Automata & Decidability of State Determinacy', in *American Control Conference*, New York City: NY, pp. 5664–5669.

Gribaudo, M., Sereno, M., Horvath, A., and Bobbio, A. (2001), 'Fluid Stochastic Petri Nets Augmented with Flush-out Arcs: Modelling and Analysis,' *Discrete Event Dynamic Systems*, 11, 97–117.

Hopcroft, J.E., Motwani, R., and Ullman, J.D. (2001), *Introduction to Automata Theory, Languages, and Computation* (2nd ed.), Boston, MA: Addison-Wesley.

Jiang, S., Kumar, R., and Garcia, H. (2003a), 'Diagnosis of Repeated Failures in Discrete Event Systems,' *IEEE Transactions on Robotics and Automation*, 19, 301–323.

Jiang, S., Kumar, R., and Garcia, H. (2003b), 'Optimal Sensor Selection for Discrete Event Systems Under Partial Observation,' *IEEE Transactions on Automatic Control*, 48, 369–381.

Leeuwen, J.V. (1990), *Handbook of Theoretical Computer Sc.: Formal Models and Semantics* (Vol. B), Cambridge, MA: Elsevier.

Lin, F., and Wonham, W.M. (1988a), 'Decentralised Control and Coordination of Discrete Event Systems with Partial Observation,' *Inf. Sci.*, 44, 199–224.

Lin, F., and Wonham, W.M. (1988b), 'On Observability of Discrete-event Systems,' *Inf. Sci.*, 44, 173–198.

Moody, J.O., and Antsaklis, P.J. (1998), *Supervisory Control of Discrete Event Systems using Petri Nets*, Boston, MA: Kluwer Academic.

Ramadge, P.J., and Wonham, W.M. (1987), 'Supervisory Control of a Class of Discrete Event Processes,' *SIAM Journal of Control and Optimization*, 25, 206–230.

Ray, A. (2005), 'Signed Real Measure of Regular Languages for Discrete-event Supervisory control,' *International Journal of Control*, 78, 949–967.

Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., and Teneketzis, D. (1995), 'Diagnosability of Discrete Event System,' *IEEE Transactions on Automatic Control*, 40, 1555–1575.

Wong, K. (1998), 'On the Complexity of Projections of Discrete-event Systems,' *IEEE Workshop on Discrete Event Systems*, Cagliari, Italy, 201–208.

Wong, K., and Wonham, W. (2004), 'The Computation of Observers in Discrete-event Systems,' *Discrete Event Dynamic Systems*, 14, 55–107.

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