7^e *Conférence Internationale de MOdélisation et SIMulation - MOSIM'08 - du* 31 *mars au* 2 *avril* 2008 *- Paris - France*
«Modélisation, Optimisation et Simulation des Systèmes : Communication, Coopération et Coordination»

A NEW ARCHITECTURE FOR DECENTRALIZED CONTROL OF DISCRETE EVENT SYSTEMS: DECIDABILITY AND SYNTHESIS ISSUES

Ahmed KHOUMSI, Hicham CHAKIB

Université de Sherbrooke, Département GEGI
2500, boulevard de l'Université
Sherbrooke, CANADA, J1K2R1
Firstname.Lastname@USherbrooke.ca

ABSTRACT : *This paper deals with decentralized control of Discrete Event Systems, where a set of supervisors take enabling/disabling decisions which can be fused for deducing actual decisions. Recently, a new control architecture has been proposed that strictly includes the solutions of all previous architectures. A new notion of* n-observability *was defined in the context of this architecture. As expected, this new* n*-observability is not preserved under union of languages.*
In the context of the above architecture, we propose a new notion of n-normality*, which is stronger than* n*-observability and preserved under union of languages. We show that these new notions of* n*-observability and* n*-normality are* undecidable*. Then we discuss how to deal with this undecidability in order to be able to* synthesize *the supremal* n*-normal and controllable language included in a given language.*

KEYWORDS : *Decentralized Supervisory Control, New Control Architecture,* n*-observability,* n*-normality, Undecidability, Synthesis of Supremal* n*-normal Language.*

1 INTRODUCTION

We study *decentralized supervisory control* (or more briefly, *decentralized control*) of discrete event systems (DES), where a set of supervisors cooperate in order to restrict the behaviour of a *plant* so that it respects a global *specification.* There are many prior articles studying decentralized control of DES (Cieslak, Desclaux, Fawaz & Varaiya 1988, Lin & Wonham 1988, Lin & Wonham 1990, Rudie & Wonham 1992, Rudie & Willems 1995, Prosser, Kam & Kwatny 1997, Ricker & Rudie 2000, Overkamp & van Schuppen 2001, Jiang & Kumar 2000, Yoo & Lafortune 2002, Yoo & Lafortune 2004, Ricker & Rudie 2003, Kumar & Takai 2005). Several decentralized architectures have been proposed (e.g., (Rudie & Wonham 1992, Yoo & Lafortune 2002, Yoo & Lafortune 2004, Kumar & Takai 2005)). The basic principle used in these architectures, is that each supervisor takes a control decision for each controllable event, according to its local observations. When several supervisors take decisions concurrently for the same event, a global decision is synthesized by "fusing" these local decisions.

Recently, a new control architecture has been proposed in (Khoumsi & Chakib 2007), which strictly includes the solutions of all the previous architectures. In contrast with the latter, the architecture of (Khoumsi & Chakib 2007) is based on the following principle:

A supervisor takes an enabling/disabling decision iff *it is sure that this is the right decision which can be applied to the plant. Otherwise, the supervisor transmits its local information to the fusion system. Consequently, when no supervisor can take a decision, the fusion system combines the elements of information transmitted by the supervisors in order to synthesize a decision.*

The approach of (Khoumsi & Chakib 2007) was motivated by the following point of view: *When a supervisor is not sure of its decision, then instead of transmitting this decision (which can be seen as a coarse information), it may transmit a richer information. The latter can for example be the event sequence the supervisor has observed or some related information.*

As with previous decentralized architectures, the authors of (Khoumsi & Chakib 2007) define a new notion of n-observability[1] in the context of their proposed architecture. As expected, this new n-observability has the same main properties as its previous versions of the previous architectures. More precisely, n-observability is preserved under intersec-

---

[1] Where n is the number of supervisors.

tion of prefix-closed languages and is *not* preserved under union of languages.

The organization of this paper is as follows, where Sections 3 and 4 correspond to our contributions:

1. In Section 2, we present the control architecture of (Khoumsi & Chakib 2007) and the corresponding notion of n-observability with its main properties.
2. Section 3 presents a new notion of n-normality, which is stronger than n-observability and preserved under union of languages (contrary to n-observability). Then, we define the supremal n-normal language included in a specification $\mathcal{K}$, which we denote $\mathcal{K}^{\uparrow N}$. n-normality with its main properties and $\mathcal{K}^{\uparrow N}$ are clarified by a few illustrative examples.
3. In Section 4, we show that these new notions of n-observability and n-normality are *undecidable*. We also discuss how to tackle this undecidability for being able to *synthesize* $\mathcal{K}^{\uparrow N}$. We also show how to synthesize $\mathcal{K}^{\uparrow CN}$, the supremal controllable and n-normal language included in $\mathcal{K}$.
4. And in Section 5, we conclude by recapitulating our contributions and proposing relevant future work.

2 CONTROL ARCHITECTURE OF (Khoumsi & Chakib 2007)

The authors of (Khoumsi & Chakib 2007) propose a new architecture for decentralized control and show that it strictly includes the solutions obtained by all the previous architectures for decentralized control. Like previous architectures, the architecture of (Khoumsi & Chakib 2007) consists of two parts: *a set of supervisors* and *a fusion system*. Let us see how the supervisors and the fusion system behave with the objective to control a plant so that it conforms to a specification. For simplicity, we will consider uniquely controllable events. Uncontrollable events are implicitly assumed always enabled.

In the sequel, G is a plant generating the prefix-closed language $\mathcal{L}(G)$ and the marked language $\mathcal{L}_m(G)$, and we consider a specification generating the language $\mathcal{K} \subseteq \mathcal{L}_m(G)$. We write $\overline{L}$ for the prefix-closure of a language L.

2.1 The Supervisors

The control architecture contains n (> 1) supervisors Sup_i, $i = 1, \cdots n$. Each Sup_i observes the plant through a projection mapping P_i which results in the observation of only a part $\Sigma_{o,i}$ of the events of the plant. And each Sup_i can control only a part $\Sigma_{c,i}$ of the events of the plant. Collectively, the supervisors observe $\Sigma_o = \Sigma_{o,1} \cup \cdots \Sigma_{o,n}$ and can control $\Sigma_c = \Sigma_{c,1} \cup \cdots \Sigma_{c,n}$. If Σ is the alphabet of the plant, we define $\Sigma_{uo} = \Sigma \setminus \Sigma_o$ and $\Sigma_{uc} = \Sigma \setminus \Sigma_c$, the sets of unobservable and uncontrollable events, respectively.

We assume that each Sup_i knows $\mathcal{L}(G), \mathcal{L}_m(G)$ and $\mathcal{K}$. Sup_i observes continuously (and partially) the plant. For every $\sigma \in \Sigma_{c,i}$ and every observed event sequence $\mu_i \in P_i(\overline{\mathcal{K}})$, Sup_i computes the following two languages:

$$\left\{ \begin{array}{ll} REJ^{\mu_i}_{i,\sigma} = & (P_i^{-1}(\mu_i) \cap \overline{\mathcal{K}})\sigma \cap (\mathcal{L}(G) \setminus \overline{\mathcal{K}}) \\ ACC^{\mu_i}_{i,\sigma} = & P_i^{-1}(\mu_i)\sigma \cap \overline{\mathcal{K}} \end{array} \right\} \quad (1)$$

Note that $ACC^{\mu_i}_{i,\sigma} \subseteq \overline{\mathcal{K}}$ and $REJ^{\mu_i}_{i,\sigma} \subseteq \mathcal{L}(G) \setminus \overline{\mathcal{K}}$. The interpretation of these two languages is that:

- If σ is accepted by $\overline{\mathcal{K}}$, then σ leads to a sequence of $ACC^{\mu_i}_{i,\sigma}$.
- If σ is accepted by $\mathcal{L}(G) \setminus \overline{\mathcal{K}}$ (we will also say: σ is *rejected* by $\overline{\mathcal{K}}$) then σ leads to a sequence of $REJ^{\mu_i}_{i,\sigma}$.

Therefore, Sup_i generates one of the following three types of outputs, for every $\sigma \in \Sigma_{c,i}$:

1. When $REJ^{\mu_i}_{i,\sigma} = \emptyset$, Sup_i deduces with certainty that σ is accepted by $\overline{\mathcal{K}}$ if it is accepted by $\mathcal{L}(G)$, and thus, its output is "Enable σ".
2. When $ACC^{\mu_i}_{i,\sigma} = \emptyset$, Sup_i deduces with certainty that σ is rejected by $\overline{\mathcal{K}}$, and thus, its output is "Disable σ".
3. When $ACC^{\mu_i}_{i,\sigma} \neq \emptyset$ and $REJ^{\mu_i}_{i,\sigma} \neq \emptyset$, Sup_i cannot determine with certainty if σ is accepted or not by $\overline{\mathcal{K}}$. In this case, its output is the pair $(ACC^{\mu_i}_{i,\sigma}; REJ^{\mu_i}_{i,\sigma}) \subseteq \overline{\mathcal{K}} \times (\mathcal{L}(G) \setminus \overline{\mathcal{K}})$.

Although the two situations $REJ^{\mu_i}_{i,\sigma} = \emptyset$ and $ACC^{\mu_i}_{i,\sigma} = \emptyset$ generate contradictory decisions (*En* and *Dis*), the situation "$ACC^{\mu_i}_{i,\sigma} = REJ^{\mu_i}_{i,\sigma} = \emptyset$" is not problematic. In fact, this situation occurs only if $(P_i^{-1}(\mu_i) \cap \overline{\mathcal{K}})\sigma \cap \mathcal{L}(G) = \emptyset$, that is, when σ is *not* accepted by $\mathcal{L}(G)$. In this case, Sup_i knows that its decision does not matter, and thus, can take any of the two decisions (enable or disable σ).

For every Sup_i and $\sigma \in \Sigma_{c,i}$, let $Sup_{i,\sigma}$ be a function that, to every observed event sequence $\mu_i \in P_i(\overline{\mathcal{K}})$, associates the corresponding output of Sup_i. Let *En* and *Dis* denote "Enable" and "Disable". That is, $Sup_{i,\sigma}(\mu_i) = En$ (resp. *Dis*) means: σ is enabled (resp. disabled) by Sup_i after the observation of μ_i. Formally: $\forall i \in \{1, \cdots, n\}, \forall \sigma \in \Sigma_{c,i}$,

$$Sup_{i,\sigma} : P_i(\overline{\mathcal{K}}) \rightarrow \{En, Dis\} \cup (2^{\overline{\mathcal{K}}} \times 2^{\mathcal{L}(G)\setminus\overline{\mathcal{K}}}).$$

More precisely, we have for $\mu_i \in P_i(\overline{\mathcal{K}})$:

$$Sup_{i,\sigma}(\mu_i) = \left\{ \begin{array}{ll} En, & \text{if } REJ_{i,\sigma}^{\mu_i} = \emptyset \\ Dis, & \text{if } ACC_{i,\sigma}^{\mu_i} = \emptyset \\ (ACC_{i,\sigma}^{\mu_i}; REJ_{i,\sigma}^{\mu_i}), & \text{otherwise} \end{array} \right\} \quad (2)$$

As we will see in the following subsection 2.2, the outputs of the supervisors are sent to the fusion system.

2.2 The Fusion System

For every event $\sigma \in \Sigma_c$, let Ind_σ be the set of indices i such that $\sigma \in \Sigma_{c,i}$. The module $Fuse_\sigma$ can be conceptually seen as a function that, to a combination of outputs of the Sup_i such that $i \in Ind_\sigma$, associates an enabling/disabling decision on σ. Formally: $\forall\sigma \in \Sigma_c, Fuse_\sigma : \prod_{i\in Ind_\sigma} Sup_{i,\sigma} \rightarrow \{En, Dis\}$.

More precisely, we have:

$$Fuse_\sigma(\prod_{i\in Ind_\sigma} Sup_{i,\sigma}(\mu_i)) = \left\{ \begin{array}{ll} En, & \text{if } (\exists i \in Ind_\sigma \text{ s.t. } Sup_{i,\sigma}(\mu_i) = En) \\ & \quad \vee (\bigcap_{i\in Ind_\sigma} REJ_{i,\sigma}^{\mu_i} = \emptyset) \\ Dis, & \text{if } (\exists i \in Ind_\sigma \text{ s.t. } Sup_{i,\sigma}(\mu_i) = Dis) \\ & \quad \vee (\bigcap_{i\in Ind_\sigma} ACC_{i,\sigma}^{\mu_i} = \emptyset) \\ Any, & \text{otherwise} \end{array} \right\} \quad (3)$$

Note that in Eq. 3, each of *En* and *Dis* has two conditions. Here are some explanations related to this equation:

- The first condition of *En* (resp. *Dis*) means that the module $Fuse_\sigma$ has received an *En* (resp. *Dis*) from at least one Sup_i. In this case, $Fuse_\sigma$ applies this decision to the plant. Notice that $Fuse_\sigma$ may receive two conflicting decisions *En* and *Dis* only if σ is *not* accepted by $\mathcal{L}(G)$. In this case, $Fuse_\sigma$ knows that its decision does not matter, and thus, can apply any of the two decisions *En* or *Dis* (see our discussion on $ACC_{i,\sigma}^{\mu_i} = REJ_{i,\sigma}^{\mu_i} = \emptyset$ in Subsection 2.1).

- The second conditions of *En* and *Dis* occur when $Fuse_\sigma$ receives no decision from the supervisors. Instead, it receives the sets $ACC_{i,\sigma}^{\mu_i}$ and $REJ_{i,\sigma}^{\mu_i}$, for $i \in Ind_\sigma$. $Fuse_\sigma$ computes the intersections $\bigcap_{i\in Ind_\sigma} REJ_{i,\sigma}^{\mu_i}$ and $\bigcap_{i\in Ind_\sigma} ACC_{i,\sigma}^{\mu_i}$, which are in some sense, refinements of the information elements received from the supervisors. The interpretation of $\bigcap_{i\in Ind_\sigma} REJ_{i,\sigma}^{\mu_i}$ and $\bigcap_{i\in Ind_\sigma} ACC_{i,\sigma}^{\mu_i}$ is that:
  - If σ is accepted by $\overline{\mathcal{K}}$, then it leads to a sequence of $\bigcap_{i\in Ind_\sigma} ACC_{i,\sigma}^{\mu_i}$. Therefore, $\bigcap_{i\in Ind_\sigma} ACC_{i,\sigma}^{\mu_i} = \emptyset$ implies that σ is certainly rejected by $\overline{\mathcal{K}}$, and thus, is disabled by $Fuse_\sigma$.
  - If σ is accepted by $\mathcal{L}(G) \setminus \overline{\mathcal{K}}$, then it leads to a sequence of $\bigcap_{i\in Ind_\sigma} REJ_{i,\sigma}^{\mu_i}$. Therefore, $\bigcap_{i\in Ind_\sigma} REJ_{i,\sigma}^{\mu_i} = \emptyset$ implies that σ is certainly accepted by $\overline{\mathcal{K}}$ if it is accepted by $\mathcal{L}(G)$, and thus, is enabled by $Fuse_\sigma$.

- The situation *Any* occurs when both $\bigcap_{i\in Ind_\sigma} ACC_{i,\sigma}^{\mu_i}$ and $\bigcap_{i\in Ind_\sigma} REJ_{i,\sigma}^{\mu_i}$ are not empty, that is, when $Fuse_\sigma$ has not enough information to decide. The occurrence of *Any* is not a problem when σ is not accepted by $\mathcal{L}(G)$, because the decision of $Fuse_\sigma$ does not matter in such a case.

  The occurrence of *Any* when σ is accepted by $\mathcal{L}(G)$, means that $\mathcal{K}$ is not achievable, that is, the control system cannot force the plant to conform to $\mathcal{K}$.

- For $\bigcap_{i\in Ind_\sigma} ACC_{i,\sigma}^{\mu_i} = \bigcap_{i\in Ind_\sigma} REJ_{i,\sigma}^{\mu_i} = \emptyset$, we can make similar comments as those already made for $REJ_{i,\sigma}^{\mu_i} = ACC_{i,\sigma}^{\mu_i} = \emptyset$ (in Subsection 2.1). That is, the situation $\bigcap_{i\in Ind_\sigma} ACC_{i,\sigma}^{\mu_i} = \bigcap_{i\in Ind_\sigma} REJ_{i,\sigma}^{\mu_i} = \emptyset$ occurs only if σ is *not* accepted by $\mathcal{L}(G)$. In this case, $Fuse_\sigma$ knows that its decision does not matter, and thus, can take any of the two decisions *En* or *Dis*.

The control architecture of (Khoumsi & Chakib 2007) is illustrated in Figure 1 for a given $\sigma \in \Sigma_c$.

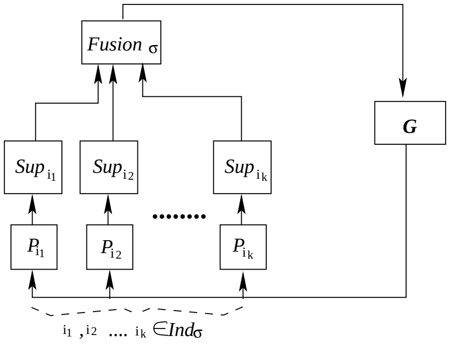

Figure 1: The control architecture of (Khoumsi & Chakib 2007)

2.3 Independency of the Fusion System

Although the architecture of (Khoumsi & Chakib

2007) generalizes the previous architectures, it keeps the following important advantage:

The essential operations of a fusion module $Fuse_\sigma$ are the computations of: $(\bigcap_{i\in Ind_\sigma} ACC^{\mu_i}_{i,\sigma})$ and $(\bigcap_{i\in Ind_\sigma} REJ^{\mu_i}_{i,\sigma})$. That is, $Fuse_\sigma$ computes *intersections* of languages received from the supervisors. Therefore, the fusion modules are *independent* of $\mathcal{L}(G), \mathcal{L}_m(G)$ and $\mathcal{K}$.

2.4 Existence Results

The notions of *n-observability* and *feasibility* have been defined in (Khoumsi & Chakib 2007).

Definition 2.1 $\mathcal{K} \subseteq \mathcal{L}_m(G)$ *is said n-observable w.r.t* $\mathcal{L}(G), \Sigma_{o,1}, \Sigma_{c,1}, \cdots, \Sigma_{o,n}, \Sigma_{c,n}$ iff $\forall \lambda, \mu \in \overline{\mathcal{K}}, \forall \sigma \in \Sigma_c$ *s.t.*
$(\lambda\sigma, \mu\sigma \in \mathcal{L}(G)) \wedge (P_i(\lambda) = P_i(\mu)\ \forall i \in Ind_\sigma)$:
$\lambda\sigma \in \overline{\mathcal{K}} \Leftrightarrow \mu\sigma \in \overline{\mathcal{K}}$.

Let us explain the intuition of n-observability of $\mathcal{K}$. We consider a controllable event σ and two event sequences λ and μ of $\overline{\mathcal{K}}$ that are undistinguished by the supervisors that control σ. Let us assume that the plant permits σ to occur after both λ and μ. n-observability of $\mathcal{K}$ guarantees that $\overline{\mathcal{K}}$: either permits σ to occur after both λ and μ, or forbids σ to occur after them.

In § 2.1 and 2.2, we explained how the *global supervisor* (consisting of the supervisors and the fusion modules), denoted SUP, interacts with G so that it conforms to $\overline{\mathcal{K}}$.

Definition 2.2 *SUP is said* feasible *w.r.t* $\overline{\mathcal{K}}, \mathcal{L}(G), \Sigma_{o,1}, \Sigma_{c,1}, \cdots, \Sigma_{o,n}, \Sigma_{c,n}$ iff
$\forall \lambda \in \overline{\mathcal{K}}, \forall \sigma \in \Sigma_c$ *such that* $\lambda\sigma \in \mathcal{L}(G)$:
$(\bigcap_{i\in Ind_\sigma} ACC^{P_i(\lambda)}_{i,\sigma} = \emptyset) \vee (\bigcap_{i\in Ind_\sigma} REJ^{P_i(\lambda)}_{i,\sigma} = \emptyset)$.

Intuitively, feasibility of SUP means that for every controllable event σ, the situation *Any* (see Subsection 2.2) never occurs when σ is accepted by $\mathcal{L}(G)$. Recall that the situation *Any* occurs when $Fuse_\sigma$ has not enough information to take an enabling/disabling decision on σ. The situation *Any* is a problem only if it occurs when σ is accepted by $\mathcal{L}(G)$, because the decision of $Fuse_\sigma$ does not matter when σ is not accepted by $\mathcal{L}(G)$.

Here is a proposition that states the equivalence between n-observability and feasibility:

Proposition 2.1 [2] *Consider a nonempty* $\mathcal{K} \subseteq \mathcal{L}_m(G)$. *The following two statements are* equivalent*:*
- SUP is feasible w.r.t $\overline{\mathcal{K}}, \mathcal{L}(G), \Sigma_{o,i}, \Sigma_{c,i},\ i = 1, \cdots, n$.
- $\mathcal{K}$ *is n-observable w.r.t* $\mathcal{L}(G), \Sigma_{o,i}, \Sigma_{c,i},\ i = 1, \cdots, n$.

When SUP is feasible, it can be seen as a function $SUP : \mathcal{L}(G) \rightarrow 2^\Sigma$ which associates to each sequence $\lambda \in \mathcal{L}(G)$ the set of enabled events. Formally: $SUP(\lambda) = \{\sigma \in \Sigma_c \mid Fuse_\sigma(\prod_{i\in Ind_\sigma} Sup_{i,\sigma}(P_i(\lambda))) = En\} \cup \Sigma_{uc}$.

When SUP is feasible, we define $\mathcal{L}(SUP/G)$ as being the prefix-closed language generated by the plant under the control of SUP. Formally:

- $\varepsilon \in \mathcal{L}(SUP/G)$ (ε is the empty event sequence)
- $[(\lambda \in \mathcal{L}(SUP/G)) \wedge (\lambda\sigma \in \mathcal{L}(G)) \wedge (\sigma \in SUP(\lambda))] \Leftrightarrow \lambda\sigma \in \mathcal{L}(SUP/G)$.

Let $\mathcal{L}_m(SUP/G) = \mathcal{L}(SUP/G) \cap \mathcal{L}_m(G)$ be the marked language corresponding to $\mathcal{L}(SUP/G)$. SUP is said nonblocking *iff* $\overline{\mathcal{L}_m(SUP/G)} = \mathcal{L}(SUP/G)$.

Theorem 2.1 [3] *Consider a nonempty* $\mathcal{K} \subseteq \mathcal{L}_m(G)$.
SUP satisfies C1 to C4 iff $\mathcal{K}$ *satisfies K1 to K3:*
C1: SUP is non blocking.
C2: SUP is feasible w.r.t $\overline{\mathcal{K}}, \mathcal{L}(G), \Sigma_{o,i}, \Sigma_{c,i},\ i = 1, \cdots, n$.
C3: $\mathcal{L}(SUP/G) = \overline{\mathcal{K}}$.
C4: $\mathcal{L}_m(SUP/G) = \mathcal{K}$.
K1: $\mathcal{K}$ *is controllable w.r.t* $\mathcal{L}(G)$ *and* Σ_{uc}.
K2: $\mathcal{K}$ *is n-observable w.r.t* $\mathcal{L}(G), \Sigma_{o,i}, \Sigma_{c,i},\ i = 1, \cdots, n$.
K3: $\mathcal{K}$ *is* $\mathcal{L}_m(G)$*-closed, that is,* $\mathcal{K} = \overline{\mathcal{K}} \cap \mathcal{L}_m(G)$.

Theorem 2.1 means that every nonempty $\mathcal{K} \subseteq \mathcal{L}_m(G)$ respecting K1 to K3 can be achieved by controlling the plant G using SUP. When the languages are prefix-closed, we can omit C1, C4 and K3.

3 *N*-NORMALITY

3.1 Definition and Properties of n-normality

Since n-observability is not preserved under union of languages, let us define a new notion of n-normality, which is stronger than n-observability and is preserved under union of languages.

Definition 3.1 $\mathcal{K} \subseteq \mathcal{L}_m(G)$ *is n-normal w.r.t* $\mathcal{L}(G), \Sigma_{o,1}, \Sigma_{c,1}, \cdots, \Sigma_{o,n}, \Sigma_{c,n}$ iff $\forall \lambda \in \overline{\mathcal{K}}, \forall \mu \in \mathcal{L}(G), \forall \sigma \in \Sigma_c$ *s.t.* $(\lambda\sigma, \mu\sigma \in \mathcal{L}(G)) \wedge (P_i(\lambda) = P_i(\mu)\ \forall i \in Ind_\sigma)$:
$\lambda\sigma \in \overline{\mathcal{K}} \Rightarrow \mu\sigma \in \overline{\mathcal{K}}$.

---

[2] This proposition corresponds to Proposition 4.1 of (Khoumsi & Chakib 2007).

[3] This theorem corresponds to Theorem 4.2 of (Khoumsi & Chakib 2007).

For brevity, in the sequel we will omit the expression "w.r.t $\mathcal{L}(G), \Sigma_{o,1}, \Sigma_{c,1}, \cdots, \Sigma_{o,n}, \Sigma_{c,n}$" related to n-observability and n-normality. Here are a few notations, where $\mathcal{K} \in \mathcal{L}_m(G)$:

Notations 3.1 *($2^{\mathcal{K}}$, $\mathcal{O}$, $\mathcal{O}_{\mathcal{K}}$, $\mathcal{N}$, $\mathcal{N}_{\mathcal{K}}$)*

- $2^{\mathcal{K}}$ *is the set of subsets of a language* $\mathcal{K}$*.*
- $\mathcal{O}$ *is the set of* n*-observable languages.*
- $\mathcal{O}_{\mathcal{K}}$ *is the set of* n*-observable languages which are included in* $\mathcal{K}$*. Formally:* $\mathcal{O}_{\mathcal{K}} = \mathcal{O} \cap 2^{\mathcal{K}}$*.*
- $\mathcal{N}$ *is the set of* n*-normal languages.*
- $\mathcal{N}_{\mathcal{K}}$ *is the set of* n*-normal languages which are included in* $\mathcal{K}$*. Formally:* $\mathcal{N}_{\mathcal{K}} = \mathcal{N} \cap 2^{\mathcal{K}}$*.*

An important property linking n-normality and n-observability, is that the former guarantees the latter. That is, if $A \subseteq \mathcal{L}_m(G)$ is n-normal, then A is n-observable. Note that the converse is not guaranteed. This is stated by the following proposition:

Proposition 3.1 $\mathcal{N} \subseteq \mathcal{O}$*.*

The advantage of n-normality is that it is preserved under union of languages (contrary to n-observability). That is, the union of n-normal languages is n-normal. This is stated by the following proposition:

Proposition 3.2 $(A, B \in \mathcal{N}) \Rightarrow (A \cup B \in \mathcal{N})$*.*

Note that propositions 3.1 and 3.2 also hold if we replace $\mathcal{N}$ and $\mathcal{O}$ by $\mathcal{N}_{\mathcal{K}}$ and $\mathcal{O}_{\mathcal{K}}$, respectively.

Since n-normality is preserved under union of languages, there exists a unique supremal n-normal language which is included in $\mathcal{K}$; let us denote it by $\mathcal{K}^{\uparrow N}$. The latter is the union of all n-normal languages included in $\mathcal{K}$. Formally:

$$\mathcal{K}^{\uparrow N} = \bigcup_{X \in \mathcal{N}_{\mathcal{K}}} X$$

For the sake of brevity, we will use the following definition:

Definition 3.2 *Given a controllable event* $\sigma \in \Sigma_c$*, two distinct sequences* λ *and* μ *are said* σ*-undistinguishable iff* λ *and* μ *cannot be distinguished by the supervisors that control* σ*. Formally:* $P_i(\lambda) = P_i(\mu)\ \forall i \in Ind_{\sigma}$*.*

3.2 Example illustrating $A \in \mathcal{O} \setminus \mathcal{N}$

In order to clarify the difference between n-observability and n-normality, let us present an example of language A which is n-observable and *not* n-normal. We consider the plant G modeled by the finite state automaton (FSA) of Figure 2 defined over the alphabet $\Sigma = \{a_1, a_2, \sigma, \alpha, \beta\}$, where $\Sigma_{uo} = \Sigma_c = \Sigma_{c,1} = \Sigma_{c,2} = \{\sigma, \alpha, \beta\}$, $\Sigma_{o,1} = \{a_1\}$, $\Sigma_{o,2} = \{a_2\}$, and $\Sigma_{uc} = \{a_1, a_2\}$. We assume the plant prefix-closed, that is, $\mathcal{L}_m(G) = \mathcal{L}(G)$, i.e., all states are marked.

We consider the prefix-closed language A obtained from the plant by forbidding α. That is, A is obtained from $\mathcal{L}(G)$ by removing the dotted part (i.e., states 6, 7 and 8). Let us show that A is n-observable and not n-normal.

The property of n-observability of A has to be checked for every controllable event $\sigma \in \Sigma_c$ and every pair of σ-undistinguishable sequences λ and μ of $\overline{A}$ such that $\mathcal{L}(G)$ permits σ to occur after both λ and μ. Since such a situation does not occur in A, we deduce that A is n-observable.

The property of n-normality of A may seem quite similar to n-observability. The only difference between the two notions, is that in n-normality, one of the two σ-indistinguishable sequences can be in $\mathcal{L}(G)$, instead of being constrained to be in $\overline{A}$. This situation holds uniquely for the two sequences $\lambda = a_1\beta a_2 \in \overline{A}$ and $\mu = a_1\alpha a_2 \in \mathcal{L}(G) \setminus \overline{A}$, that lead to states 4 and 7 respectively. Indeed, $P_1(\lambda) = P_1(\mu) = a_1$ and $P_2(\lambda) = P_2(\mu) = a_2$, and $\mathcal{L}(G)$ permits σ to occur after both λ and μ, i.e., in both states 4 and 7. A is not n-normal because $\lambda\sigma \in \overline{A}$ (leads to state 5) and $\mu\sigma \notin \overline{A}$ (leads to state 8).

Note that we obtain a n-normal behavior if we remove state 5 from A. This is actually the supremal $A^{\uparrow N}$.

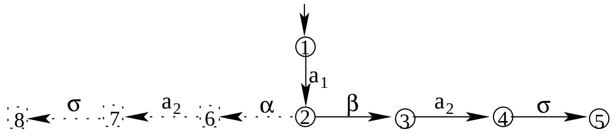

Figure 2: Example illustrating $A \in \mathcal{O} \setminus \mathcal{N}$

3.3 Example Illustrating $(A, B \in \mathcal{N}) \Rightarrow (A \cup B \in \mathcal{N})$

Let us present an example which illustrates the preservation of n-normality under union of languages. We consider the prefix-closed plant G of the previous example. Therefore, $\Sigma = \{a_1, a_2, \sigma, \alpha, \beta\}$, $\Sigma_{uo} = \Sigma_c = \Sigma_{c,1} = \Sigma_{c,2} = \{\sigma, \alpha, \beta\}$, $\Sigma_{o,1} = \{a_1\}$, $\Sigma_{o,2} = \{a_2\}$, and $\Sigma_{uc} = \{a_1, a_2\}$.

We consider the two prefix-closed specifications A and B of Figure 3. Their union $A \cup B$ is represented in the same figure. In each language X (A, B or $A \cup B$), the dotted part represents what must be removed from the plant for obtaining X.

Recall that the property of n-normality of a language X has to be checked for every $\sigma \in \Sigma_c$ and every pair of σ-undistinguishable sequences $\lambda \in \overline{X}$ and $\mu \in \mathcal{L}(G)$ such that $\mathcal{L}(G)$ permits σ to occur after both λ and μ. In A, this situation holds uniquely for $\lambda = a_1\beta a_2 \in \overline{A}$ and $\mu = a_1\alpha a_2 \in \mathcal{L}(G) \setminus \overline{A}$, that lead to states 4 and 7 respectively and are followed by $\sigma \in \Sigma_c$. A is n-normal because both $\lambda\sigma$ and $\mu\sigma$ are not accepted by $\overline{A}$. In B, the situation holds uniquely for $\lambda = a_1\alpha a_2 \in \overline{B}$ and $\mu = a_1\beta a_2 \in \mathcal{L}(G) \setminus \overline{B}$, that lead to states 7 and 4 respectively and are followed by $\sigma \in \Sigma_c$. B is n-normal because both $\lambda\sigma$ and $\mu\sigma$ are not accepted by $\overline{B}$. Using the same reasoning, we can easily deduce that $A \cup B$ is n-normal.

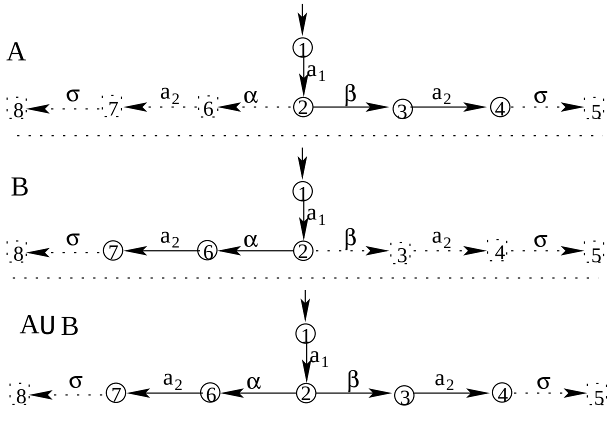

Figure 3: Example illustrating the preservation of n-normality under union of languages

4 ON UNDECIDABILITY AND SYNTHESIS

4.1 Undecidability of n-observability and n-normality

In (Tripakis 2001), a joint-observability has been defined and shown to be undecidable. From the resemblance of the joint-observability with n-observability and n-normality, we can easily state (and prove) the following Proposition:

Proposition 4.1 *n-observability and n-normality are undecidable in general.*

4.2 How to Tackle this Undecidability

For simplicity of the discussion, we suppose here that the plant and the specification are prefix-closed, i.e., $\mathcal{L}_m(G) = \mathcal{L}(G)$ and $\mathcal{K} = \overline{\mathcal{K}}$.

Note that undecidability occurs only in problems using *infinite* sets. Therefore, a simple idea that crosses the mind is to consider only *finite* sets. In our context, this means that the plant and the specification (i.e., $\mathcal{L}(G)$ and $\mathcal{K}$) are finite languages. When the latter are infinite (i.e., in the general case), an approach may consist in approximating them by finite languages. The question that arises is then: *When is it possible to approximate realistically the plant and the specification by finite languages ?* We present here a tentative to answer to this question: we propose a very simple approximation method, and then we analyze briefly when this method is realistic.

Our proposition for approximating a language A is to keep only the sequences whose lengths are lower than a fix upper bound. Let us denote by A_f the obtained finite language. In our context of control, $\mathcal{L}(G)$ and $\mathcal{K}$ are approximated by two finite languages $\mathcal{L}(G)_f$ and $\mathcal{K}_f$, by keeping only the event sequences whose lengths are lower than a given value ℓ.

This approach of approximation seems realistic if we assume that the plant is not executed indefinitely, that is, it is inevitably *stopped* after an execution whose length is lower than a given upper bound ℓ. We may also assume that the plant is inevitably *reset* (i.e., returns to its initial state[4] instead of being stopped) after an execution whose length is lower than ℓ. In the latter case, the plant is said *periodic* and we suppose that all the supervisors observe (i.e., are informed of) the reset. The value ℓ must be chosen sufficiently great to exceed the length of the longest event sequence (or path) between two passages through the initial state of the plant.

In the sequel, we assume that we are in the presence of *finite* languages $\mathcal{L}(G)$ and $\mathcal{K}$, and thus, checking n-observability and n-normality are decidable.

4.3 Fix-Point Method for Computing $\mathcal{K}^{\uparrow N}$

The principle for computing $\mathcal{K}^{\uparrow N}$ is based on the iterative execution of the following operator Ω, where $A \subseteq \mathcal{L}_m(G)$:

$$\Omega(A) = A \setminus \{\lambda\sigma x \mid \sigma \in \Sigma_c, \lambda, x \in \Sigma^*, \exists \mu\sigma \in \mathcal{L}(G) \setminus \overline{A}, \\ (P_i(\lambda) = P_i(\mu) \, \forall i \in Ind_\sigma)\}$$

The objective of the operator Ω is to remove the "problematic" sequences, that is, sequences that make A non n-normal. But $\Omega(A)$ is not necessarily n-normal, because some "good" sequences can become problematic after the removal of problematic sequences. We have the following lemmas:

Lemma 4.1 $(\mathcal{K} \in \mathcal{N}) \Leftrightarrow (\Omega(\mathcal{K}) = \mathcal{K})$.

---

[4] The initial state of the plant means the initial state of the corresponding minimal finite state automaton.

Lemma 4.2 $(A \subseteq B) \Rightarrow (\Omega(A) \subseteq \Omega(B))$

When $\mathcal{K}$ is not n-normal, let us show how $\mathcal{K}^{\uparrow N}$ can be computed by using Ω iteratively. We consider the following series $K_0, K_1, \cdots, K_i, \cdots$:

- $K_0 = \mathcal{K}$
- $K_{i+1} = \Omega(K_i)$, $\forall i \geq 0$.

Proposition 4.2 *$\exists p \geq 0$ such that $\forall q \geq p$: $K_q = K_p$. That is, Ω has a fix-point which is obtained after a finite number p of iterations.*

Proposition 4.3 *$\mathcal{K}^{\uparrow N} = K_p$. That is, $\mathcal{K}^{\uparrow N}$ is the fix-point of the operator Ω.*

4.4 Example of $\mathcal{K}^{\uparrow N}$ Synthesis

We consider the plant G modeled by the FSA of Figure 4 defined over the alphabet $\Sigma = \{a_1, a_2, \sigma, \alpha, \beta, \gamma\}$, where $\Sigma_{uo} = \Sigma_c = \Sigma_{c,1} = \Sigma_{c,2} = \{\sigma, \alpha, \beta, \gamma\}$, $\Sigma_{o,1} = \{a_1\}$, $\Sigma_{o,2} = \{a_2\}$, and $\Sigma_{uc} = \{a_1, a_2\}$. We assume the plant prefix-closed, i.e., all states are marked.

We consider the prefix-closed specification $\mathcal{K}$ obtained from the plant by forbidding α, that is, by removing the dotted part (containing states 10, 11 and 12). Let us show that $\mathcal{K}$ is not n-normal.

The property of n-normality of $\mathcal{K}$ has to be checked for every $\sigma \in \Sigma_c$ and every pair of σ-undistinguishable sequences $\lambda \in \overline{\mathcal{K}}$ and $\mu \in \mathcal{L}(G)$ which are followed by σ in $\mathcal{L}(G)$. This is for example the case for $\lambda = a_1\beta a_2 \in \overline{\mathcal{K}}$ and $\mu = a_1\alpha a_2 \in \mathcal{L}(G)$ that lead to states 4 and 11 respectively and are followed by $\sigma \in \Sigma_c$. $\mathcal{K}$ is not n-normal because $\lambda\sigma \in \overline{\mathcal{K}}$ (leads to state 7) and $\mu\sigma \notin \overline{\mathcal{K}}$ (leads to state 12).

If we apply Ω to $\mathcal{K}$ of Figure 4, we obtain $K_1 = \Omega(\mathcal{K})$ of Figure 5, which is obtained from $\mathcal{K}$ by forbidding the σ which links states 4 and 7. The dotted part of Figure 5 represents what must be removed from the plant for obtaining K_1. Let us show that this K_1 is not n-normal.

Let us consider the two undistinguishable sequences $\lambda = a_1\beta a_2 a_1 \in \overline{K_1}$ and $\mu = a_1\beta a_2\sigma a_1 \in \mathcal{L}(G)$, that lead to states 5 and 8 respectively and are followed by $\gamma \in \Sigma_c$. K_1 is not n-normal because $\lambda\gamma \in \overline{K_1}$ (leads to state 6) and $\mu\gamma \notin \overline{K_1}$ (leads to state 9).

If we apply Ω to K_1 of Figure 5, we obtain $K_2 = \Omega(K_1)$ $(= \Omega(\Omega(\mathcal{K})))$ of Figure 6. The dotted part of Figure 6 represents what must be removed from the plant for obtaining K_2. Let us show that this K_2 is n-normal.

There exist two pairs (λ, μ) such that: $\lambda \in \overline{K_2}$ and $\mu \in \mathcal{L}(G)$ are indistinguishable and $\mathcal{L}(G)$ permits the same controllable event after both λ and μ. The two pairs are:

- $\lambda = a_1\beta a_2 a_1$, $\mu = a_1\beta a_2\sigma a_1$ and $\mathcal{L}(G)$ permits the controllable event γ to occur after both λ and μ.
- $\lambda' = a_1\alpha a_2$, $\mu' = a_1\beta a_2$ and $\mathcal{L}(G)$ permits the controllable event σ to occur after both λ' and μ'.

K_2 is n-normal because in the above two situations $\lambda\gamma, \mu\gamma \notin \overline{K_2}$ and $\lambda'\sigma, \mu'\sigma \notin \overline{K_2}$.

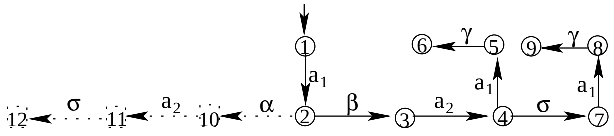

Figure 4: Example of $\mathcal{K}$ which is not n-normal

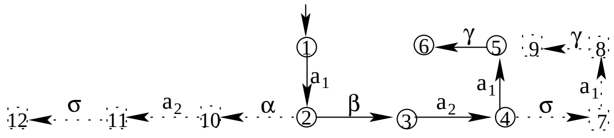

Figure 5: $K_1 = \Omega(K_0)$ for the $K_0 = \mathcal{K}$ of Figure 4

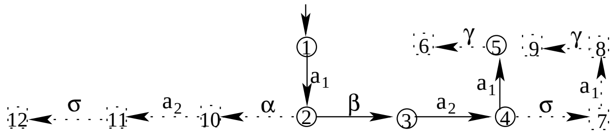

Figure 6: $K_2 = \Omega(K_1)$ $(= \Omega(\Omega(K_0)))$ for the K_1 of Figure 5

4.5 Synthesis of the Supremal Controllable and n-normal Language

We have seen in Theorem 2.1 that a language $X \in \mathcal{L}_m(G)$ is achievable by controlling the plant, *iff* X is controllable, n-observable and $\mathcal{L}_m(G)$-closed. For simplicity, let us assume the languages prefix-closed, and thus, $\mathcal{L}_m(G)$-closure is guaranteed. Therefore, the necessary and suffisant conditions for achievability become uniquely controllability and n-observability. Since n-normality guarantees n-observability, we deduce that controllability and n-normality are sufficient conditions for the achievability of $X \in \mathcal{L}_m(G)$.

It is well known that controllability is preserved under union of languages and that there exists a unique supremal controllable language which is included in a language $X \subseteq \mathcal{L}_m(G)$; let us denote it by $X^{\uparrow C}$.

Since both controllability and n-normality are preserved under union of languages, we also deduce that there exists a unique supremal controllable and n-normal language which is included in X; let us denote it by $X^{\uparrow CN}$. The latter is the union of all languages included in X which are both controllable and n-normal.

We denote by CONT and NORM the functions that compute the supremal controllable language and the supremal n-normal language, respectively. That is, if $X \subseteq \mathcal{L}_m(G)$:

- $\text{CONT}(X) = X^{\uparrow C}$,
- $\text{NORM}(X) = X^{\uparrow N}$.

Given a specification $\mathcal{K} \subseteq \mathcal{L}_m(G)$, $\mathcal{K}^{\uparrow CN}$ can be computed by using CONT and NORM iteratively and alternately, until a fix-point is reached. That is, we compute a series $K_1, K_2, \cdots$ and we stop when $K_{i+1} = K_i$. The series K_i is defined as follows:

- $K_0 = \mathcal{K}$
- $K_{2i+1} = \text{CONT}(K_{2i})$, $\forall i \geq 0$
- $K_{2i+2} = \text{NORM}(K_{2i+1})$, $\forall i \geq 0$

We obtain the same fix-point if we switch CONT and NORM, that is:

- $K_0 = \mathcal{K}$
- $K_{2i+1} = \text{NORM}(K_{2i})$, $\forall i \geq 0$
- $K_{2i+2} = \text{CONT}(K_{2i+1})$, $\forall i \geq 0$

4.6 Discussions on the computations of $\mathcal{K}^{\uparrow N}$ and $\mathcal{K}^{\uparrow CN}$

In Subsection 4.3, we have presented a function NORM that computes $\mathcal{K}^{\uparrow N}$ and is based on the iterative application of the operator Ω. Recall that the actual aim is to obtain a *n-observable* language; n-normality is used just for its convenient preservation under union of languages. Consequently, we can adopt a modified version of NORM, which we denote NORM_{obs}, where the condition that stops the iterative application of Ω is the obtention of a n-observable language (even if not n-normal), instead of a n-normal language.

For example in Section 4.4, $\mathcal{K}$ of Figure 4 is n-observable. Therefore, the two iterations for computing K_1 and K_2 of Figures 5 and 6 are not applied if we use NORM_{obs}. That is, $\text{NORM}_{\text{obs}}(\mathcal{K}) = \mathcal{K}$.

We can also replace NORM by NORM_{obs} in the function of Subsection 4.5 that applies CONT and NORM iteratively and alternately. In this case, we stop when K_i is controllable and n-observable (even if not n-normal).

5 CONCLUSION

A new control architecture has been recently proposed in (Khoumsi & Chakib 2007), which generalizes all prior architectures for decentralized control of DES. In this paper, we have first presented this new architecture and the corresponding notion of n-observability, which is not preserved under union of languages. Then, we have proposed a new notion of n-normality, which is stronger than n-observability and has the advantage to be preserved under union of languages. We have shown that these new notions of n-observability and n-normality are *undecidable* and we proposed simple sufficient conditions that make these two notions decidable. Assuming these conditions, we have developed a method for synthesizing $\mathcal{K}^{\uparrow N}$ (supremal n-normal language included in $\mathcal{K}$) and $\mathcal{K}^{\uparrow CN}$ (supremal controllable and n-normal language included in $\mathcal{K}$).

Here are a few points we plan to study in a near future:

- To evaluate rigorously the complexity for computing $\mathcal{K}^{\uparrow N}$ and $\mathcal{K}^{\uparrow CN}$.
- To tackle more efficiently the undecidability problem. More precisely, the aim is to investigate less restrictive (and less simple) conditions that make n-observability and n-normality decidable.

References

Cieslak, R., Desclaux, C., Fawaz, A. & Varaiya, P. (1988). Supervisory control of discrete event processes with partial observations, *IEEE Transactions on Automatic Control* **33**(3): 249–260.

Jiang, S. & Kumar, R. (2000). Decentralized control of discrete event systems with specializations to local control and concurrent systems, *IEEE Transactions on Systems, Man, and Cybernetics, Part B* **30**(5): 653–660.

Khoumsi, A. & Chakib, H. (2007). Decentralized supervisory control of discrete event systems: Involving the fusion systems in the decision-making, *Proc. 10th IASTED Intern. Conf. on Intelligent Systems and Control (ISC)*, Cambridge, Massachusetts, USA.

Kumar, R. & Takai, S. (2005). Inference-based ambiguity management in decentralized decision-making: Decentralized control of discrete event systems, *Conference on Decision and Control (CDC)*, Seville, Spain.

Lin, F. & Wonham, W. M. (1988). Decentralized supervisory control of discrete event systems, *Information Sciences* **44**: 199–224.

Lin, F. & Wonham, W. M. (1990). Decentralized control and coordination of discrete-event systems with partial observation, *IEEE Transactions of Automatic Control* **35**(12): 1330–1337.

Overkamp, A. & van Schuppen, J. H. (2001). Maximal solutions in decentralized supervisory control, *SIAM Journal on Control and Optimization* **39**(2): 492–511.

Prosser, J. H., Kam, M. & Kwatny, H. G. (1997). Decision fusion and supervisor synthesis in decentralized discrete-event systems, *American Control Conference (ACC)*, pp. 1313–1319.

Ricker, S. L. & Rudie, K. (2003). Knowledge is a terrible thing to waste: using inference in discrete-event control problems, *American Control Conference (ACC)*, Denver, CO, USA.

Ricker, S. & Rudie, K. (2000). Know means no: Incorporating knowledge into discrete-event control systems, *IEEE Transactions on Automatic Control* **45**(9): 1656–1668.

Rudie, K. & Willems, J. C. (1995). The computational complexity of decentralized discrete event control problems, *IEEE Transactions on Automatic Control* **40**(7): 1313–1319.

Rudie, K. & Wonham, W. M. (1992). Think globally, act locally: decentralized supervisory control, *IEEE Transactions on Automatic Control* **31**(11): 1692–1708.

Tripakis, S. (2001). Undecidable problems of decentralized control and observation, *IEEE Conference on Decision and Control (CDC)*, Orlando, FL, USA.

Yoo, T.-S. & Lafortune, S. (2002). A general architecture for decentralized supervisory control of discrete-event systems, *Discrete Event Dyna. Syst.: Theory Applicat.* **12**: 335–377.

Yoo, T.-S. & Lafortune, S. (2004). Decentralized supervisory control with conditional decisions: Supervisor existence, *IEEE Transactions on Automatic Control* **49**(11): 1886–1904.

A Proofs

A.1 Proof of Proposition 3.1

Let us first prove the following Proposition:

Proposition A.1 $\mathcal{K} \subseteq \mathcal{L}_m(G)$ *is n-normal w.r.t* $\mathcal{L}(G), \Sigma_{o,1}, \Sigma_{c,1}, \cdots, \Sigma_{o,n}, \Sigma_{c,n}$ iff $\forall\lambda\in\overline{\mathcal{K}}, \forall\mu\in\mathcal{L}(G)$, $\forall\sigma \in \Sigma_c$ *s.t.* $(\lambda\sigma, \mu\sigma \in \mathcal{L}(G)) \wedge (P_i(\lambda) = P_i(\mu)\, \forall i \in Ind_\sigma) : \lambda\sigma \in \overline{\mathcal{K}} \Leftrightarrow \mu\sigma \in \overline{\mathcal{K}}$.

Prop. A.1 is obtained from Def. 3.1 by simply replacing the implication "$\lambda\sigma \in \overline{\mathcal{K}} \Rightarrow \mu\sigma \in \overline{\mathcal{K}}$" by the equivalence "$\lambda\sigma \in \overline{\mathcal{K}} \Leftrightarrow \mu\sigma \in \overline{\mathcal{K}}$". Let us prove that this replacement is correct, that is, for a n-normal $\mathcal{K}$ the implication guarantees the equivalence. Or in other words, let us prove that "$\lambda\sigma \in \overline{\mathcal{K}} \Rightarrow \mu\sigma \in \overline{\mathcal{K}}$" guarantees the converse "$\mu\sigma \in \overline{\mathcal{K}} \Rightarrow \lambda\sigma \in \overline{\mathcal{K}}$".

We consider a n-normal $\mathcal{K} \subseteq \mathcal{L}_m(G)$, that is, from Def. 3.1: $\forall\lambda\in\overline{\mathcal{K}}, \forall\mu\in\mathcal{L}(G), \forall\sigma\in\Sigma_c$ s.t.
$(\lambda\sigma, \mu\sigma \in \mathcal{L}(G)) \wedge (P_i(\lambda) = P_i(\mu)\, \forall i \in Ind_\sigma) :$
$\lambda\sigma \in \overline{\mathcal{K}} \Rightarrow \mu\sigma \in \overline{\mathcal{K}}$.

If we assume $\mu\sigma \in \overline{\mathcal{K}}$, we deduce $\mu \in \overline{\mathcal{K}}$ and thus, the roles of λ and μ can be switched in Def. 3.1. Therefore, we obtain $\mu\sigma \in \overline{\mathcal{K}} \Rightarrow \lambda\sigma \in \overline{\mathcal{K}}$, which proves Prop. A.1.

Let us now compare Def. 2.1 and Prop. A.1 to prove that n-observability guarantees n-normality. We see that Prop. A.1 is obtained from Def. 2.1 by simply weakening the expression $\mu \in \overline{\mathcal{K}}$ into $\mu \in \mathcal{L}(G)$. Therefore, n-normality is stronger than (i.e., guarantees) n-observability.

More precisely, we consider a n-normal $A \subseteq \mathcal{L}_m(G)$, and the aim is to prove that A is necessarily n-observable. Since A is n-normal, we have from Prop. A.1: $\forall\lambda\in\overline{A}, \forall\mu\in\mathcal{L}(G), \forall\sigma\in\Sigma_c$ s.t. $(\lambda\sigma, \mu\sigma \in \mathcal{L}(G)) \wedge (P_i(\lambda) = P_i(\mu)\, \forall i \in Ind_\sigma) : (\lambda\sigma \in \overline{A} \Leftrightarrow \mu\sigma \in \overline{A})$.

The above definition of n-normality of A states a property which is satisfied for every pair of sequences $\lambda\in\overline{A}$ and $\mu\in\mathcal{L}(G)$. Since $\overline{A} \subseteq \mathcal{L}(G)$, the property also holds if $\mu\in\overline{A}$. That is: $\forall\lambda\in\overline{A}, \forall\mu\in\overline{A}, \forall\sigma\in\Sigma_c$ s.t. $(\lambda\sigma, \mu\sigma \in \mathcal{L}(G)) \wedge (P_i(\lambda) = P_i(\mu)\, \forall i \in Ind_\sigma) :$ $(\lambda\sigma \in \overline{A} \Leftrightarrow \mu\sigma \in \overline{A})$.

The latter property corresponds to the definition of n-observability of A.

A.2 Proof of Proposition 3.2

We consider two n-normal languages $A, B \subseteq \mathcal{L}_m(G)$, and the aim is to prove that $A \cup B$ is necessarily n-normal.

1. We consider $\lambda \in \overline{A \cup B}, \mu \in \mathcal{L}(G), \sigma \in \Sigma_c$ s.t. $(\lambda\sigma \in \overline{A \cup B}) \wedge (\mu\sigma \in \mathcal{L}(G)) \wedge (P_i(\lambda) = P_i(\mu)\, \forall i \in \mathit{Ind}_\sigma)$.
2. $\lambda\sigma \in \overline{A \cup B}$ of Item 1 implies: $\lambda\sigma \in \overline{A}$ or $\lambda\sigma \in \overline{B}$.
3. If in Item 2, $\lambda\sigma \in \overline{A}$, then $\lambda \in \overline{A}$. From Item 1 and n-normality of A, we deduce $\mu\sigma \in \overline{A}$.
4. If in Item 2, $\lambda\sigma \in \overline{B}$, then $\lambda \in \overline{B}$. From Item 1 and n-normality of B, we deduce $\mu\sigma \in \overline{B}$.
5. Items 3 and 4 imply $\mu\sigma \in \overline{A} \cup \overline{B}$.
6. Item 5 implies $\mu\sigma \in \overline{A \cup B}$.
7. Items 1 and 6 imply that $A \cup B$ is n-normal.

A.3 Proof of Proposition 4.1

By analogy with the joint-observability of (Tripakis 2001), we can adapt quite easily the proof of the undecidability of the joint-observability which can be found in (Tripakis 2001).

A.4 Proof of Lemma 4.1

Let us prove that $\mathcal{K}$ is n-normal *iff* $\Omega(\mathcal{K}) = \mathcal{K}$.

1. n-normality of $\mathcal{K}$ and Def. 3.1 is equivalent to :
   $\forall \lambda \in \overline{\mathcal{K}}, \forall \mu \in \mathcal{L}(G), \forall \sigma \in \Sigma_c$ s.t.
   $(\lambda\sigma, \mu\sigma \in \mathcal{L}(G)) \wedge (P_i(\lambda) = P_i(\mu)\, \forall i \in \mathit{Ind}_\sigma)$:
   $\lambda\sigma \in \overline{\mathcal{K}} \Rightarrow \mu\sigma \in \overline{\mathcal{K}}$.
2. Item 1 can be written: $\forall \lambda \in \overline{\mathcal{K}}, \forall \sigma \in \Sigma_c$ s.t. $\lambda\sigma \in \overline{\mathcal{K}}$: $\neg\exists \mu \in \mathcal{L}(G)$ s.t.
   $(\mu\sigma \in \mathcal{L}(G) \setminus \overline{\mathcal{K}}) \wedge (P_i(\lambda) = P_i(\mu)\, \forall i \in \mathit{Ind}_\sigma)$.
3. Item 2 can be written:
   $\forall \lambda \in \overline{\mathcal{K}}, \forall \sigma \in \Sigma_c, \forall x \in \Sigma^*$ s.t. $\lambda\sigma x \in \mathcal{K}$:
   $\neg\exists \mu \in \mathcal{L}(G)$ s.t.
   $(\mu\sigma \in \mathcal{L}(G) \setminus \overline{\mathcal{K}}) \wedge (P_i(\lambda) = P_i(\mu)\, \forall i \in \mathit{Ind}_\sigma)$.
4. Item 3 means $\Omega(\mathcal{K}) = \mathcal{K}$.

A.5 Proof of Lemma 4.2

Lemma 4.2 is equivalent to:

$[(A \subseteq B) \wedge (\varphi \in \Omega(A)] \Rightarrow [\varphi \in \Omega(B)]$.

1. We consider A and B such that $A \subseteq B$.
2. We consider a sequence $\varphi \in \Omega(A)$, that is:
   $\varphi \in A \setminus \{\lambda\sigma x \mid \sigma \in \Sigma_c, \lambda, x \in \Sigma^*, \exists \mu\sigma \in \mathcal{L}(G) \setminus \overline{A},$
   $(P_i(\lambda) = P_i(\mu)\, \forall i \in \mathit{Ind}_\sigma)\}$.
3. Item 2 implies:
   3.a: $\varphi \in A$ and
   3.b: If $\varphi = \lambda\sigma x$ for $\sigma \in \Sigma_c, \lambda, x \in \Sigma^*$,
   Then there exists no $\mu\sigma \in \mathcal{L}(G) \setminus \overline{A}$ such that $(P_i(\lambda) = P_i(\mu)\, \forall i \in \mathit{Ind}_\sigma)$.
4. Items 1 and 3.a imply $\varphi \in B$.
5. Item 1 implies $\mathcal{L}(G) \setminus \overline{B} \subseteq \mathcal{L}(G) \setminus \overline{A}$.
6. Items 3.b and 5 imply:
   If $\varphi = \lambda\sigma x$ for $\sigma \in \Sigma_c, \lambda, x \in \Sigma^*$,
   Then there exists no $\mu\sigma \in \mathcal{L}(G) \setminus \overline{B}$ such that $(P_i(\lambda) = P_i(\mu)\, \forall i \in \mathit{Ind}_\sigma)$
7. Items 4 and 6 imply: $\varphi \in \Omega(B)$.

A.6 Proof of Proposition 4.2

1. The operator Ω consists in removing some "problematic" sequences and is applied iteratively starting from $\mathcal{K}$. The iterative process is stopped when no sequence is removed, i.e., when a fix-point is reached.
2. $\mathcal{K}$ is supposed finite, i.e., contains a finite number of sequences)
3. Items 1 and 2 imply that the fix-point is reached after a finite number of iterations. Note that the fix-point can be empty.

A.7 Proof of Proposition 4.3

Let us show that the fix-point of Ω is $\mathcal{K}^{\uparrow N}$.

By definition, the fix-point K_p of Ω is such that : $K_p^{\uparrow N} = K_p$. From Lemma 4.1, we deduce that the fix-point K_p is n-normal.

Therefore, it remains to show that if there exists a n-normal language M such that $K_p \subseteq M \subseteq K_0$ $(= \mathcal{K})$, then $K_p = M$.

1. We consider an n-normal M such that $K_p \subseteq M \subseteq K_0$.
2. Lemma 4.1 and n-normality of K_p and M imply: $\Omega(K_p) = K_p$ and $\Omega(M) = M$.
3. Let us apply Ω p times in Item 1. From Lemma 4.2 and Item 2, we obtain $K_p \subseteq M \subseteq K_p$. Therefore, $K_p = M$.