Decentralized Supervisory Control of Discrete Event Systems: Using Multi-Decision Control as an Alternative to Inference-Based Control

Ahmed Khoumsi and Hicham Chakib
University of Sherbrooke, Dep. Elec. & Comp. Eng., Sherbrooke, Canada

Keywords: Discrete Event Systems, Multi-Decision Control, C&P-control, D&A-control, Inference-Based Control.

Abstract: Some years ago, a decentralized architecture, qualified as inference-based, has been developed for supervisory control of discrete event systems. One of its essential principles is to associate ambiguity levels to local decisions. More recently, a decentralized control architecture, qualified as multi-decision, has been developed. Its main principle is to use several decentralized control architectures in parallel. So far, multi-decision control has been mostly studied as a solution to generalize inference-based control, by using several inference-based architectures in parallel. In the present study, we regard multi-decision control with a different perspective. Instead of using multi-decision control to *generalize* inference-based control, we will rather use it as an *alternative* to inference-based control. More precisely, our objective is to avoid using inference-based architectures, by using instead several simpler architectures running in parallel.

1 INTRODUCTION

This paper is about decentralized control, where several local supervisors cooperate in order to restrict the behavior of a plant so that it respects a given specification. The authors of (Kumar and Takai, 2007) propose an interesting decentralized control approach, called inference-based control, where an ambiguity level is associated to the decision of each local supervisor. The principle is that among the decisions of the local supervisors, the effective decision which is selected is the one with the lowest ambiguity. In (Kumar and Takai, 2007), it is shown that inference-based control generalizes several previous control architectures, in particular C&P and D&A architectures (Rudie and Wonham, 1992; Yoo and Lafortune, 2002), which are the simplest interesting decentralized architectures. The authors of (Yoo and Lafortune, 2002) also combine C&P and D&A to obtain a so-called C&P∨D&A architecture. C&P, D&A and C&P∨D&A are in fact specific cases of inference-based control, where uniquely the null ambiguity level is associated to the local decisions.

Recently, a decentralized architecture for control, qualified as multi-decision, has been developed (Chakib and Khoumsi, 2011). Its principle is to use several decentralized architectures in parallel whose decisions are combined disjunctively or conjunctively. Each of the architectures in parallel can be any of the known decentralized architectures, for example C&P∨D&A or inference-based. The obtained multi-decision architecture generalizes all the architectures in parallel, in the sense that it permits to achieve, not only all the languages achievable by each of the architectures in parallel, but also languages achievable by none of the architectures in parallel. In (Chakib and Khoumsi, 2011), multi-decision control is studied as a solution to generalize inference-based control, by using inference-based architectures running in parallel.

Compared to the multi-decision control, the inference-based control has the advantage that its architecture does not depend on the structures of the automata modeling the plant and the specification. But on the other side, the inference-based architecture is relatively complex. So a question that arises is: *Is it possible to avoid using inference-based control, by using instead multi-decision with several simple control architectures running in parallel, without losing in generality ?* Our study is an attempt to answer that question. The simple architectures in parallel we will consider are qualified as C&P∨D&A, which are the simplest relevant decentralized architectures known in the literature.

The rest of the paper is organized a follows. In Section 2, we introduce decentralized control with an emphasis on pertinent architectures for our study, namely C&P∨D&A and inference-based controls.

Khoumsi, A. and Chakib, H.
Decentralized Supervisory Control of Discrete Event Systems: Using Multi-Decision Control as an Alternative to Inference-Based Control.
DOI: 10.5220/0005990102130220
In *Proceedings of the 13th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2016) - Volume 1*, pages 213-220
ISBN: 978-989-758-198-4

Section 3 presents multi-decision control with an emphasis on pertinent results obtained in previous research. In Section 4, we develop a procedure that constructs a multi-decision architecture consisting uniquely of C&P∨D&A architectures in parallel, instead of inference-based architectures. Section 5 concludes our study.

2 DECENTRALIZED CONTROL

Let *alphabet* denote a finite set of events, *trace* denote a finite sequence of events, and *language* denote a set of traces. A trace λ is said a prefix of a trace μ if there exists a trace α such that $\mu = \lambda\alpha$. $\overline{L}$ denotes the set of prefixes of a language L. $L \setminus K$ denotes the language L without the traces of the language K. Let $P_i(X)$ and $P_i^{-1}(X)$ denote the usual projection and inverse projection of a language or an automaton X.

2.1 Supervisory Control

Supervisory control (or more succinctly: control) consists in forcing a discrete event system (DES), called plant, to achieve the language of a given specification. More precisely, the objective is to restrict the behavior of the plant so that it executes uniquely traces accepted by the specification. The plant is modeled by a finite state automaton (FSA) $\mathcal{G} = (Q, \Sigma, \delta, q_0, Q_m)$, where Q is a finite set of states, Σ is an alphabet, the transitions are specified by a partial function $\delta : Q \times \Sigma \to Q$, $q_0 \in Q$ is the initial state, and $Q_m \subseteq Q$ is the set of marked states. We denote by L and L_m the generated and marked languages of $\mathcal{G}$, respectively. We have, $\overline{L_m} \subseteq L$. Intuitively, L (resp. L_m) contains the traces of $\mathcal{G}$ starting in q_0 and reaching Q (resp. Q_m). In the same way, we consider a specification modeled by a trim FSA $\mathcal{K} = (R, \Sigma, \xi, r_0, R_m)$, and we denote by K the marked language of $\mathcal{K}$. Since $\mathcal{K}$ is trim, its generated language is $\overline{K}$. We have the following notion of L_m-closure which is fundamental in control:

Definition 2.1. *K is said L_m-closed if $K = \overline{K} \cap L_m$.*

The alphabet Σ is partitioned into Σ_c and Σ_{uc}, the set of controllable and uncontrollable events, respectively. We define the following languages $\mathcal{E}_\sigma$ and $\mathcal{D}_\sigma$:

Definition 2.2. *For every event $\sigma \in \Sigma$, $\mathcal{E}_\sigma = \{\lambda \in \overline{K} \,|\, \lambda\sigma \in \overline{K}\}$ and $\mathcal{D}_\sigma = \{\lambda \in \overline{K} \,|\, \lambda\sigma \in L \setminus \overline{K}\}$. Intuitively, $\mathcal{E}_\sigma$ is the set of traces of the specification after which σ is accepted by both the plant and the specification, while $\mathcal{D}_\sigma$ is the set of traces of the specification after which σ is accepted only by the plant.*

The control is realized by the means of a supervisor *SUP* that observes continuously the evolution of the plant in order to decide to enable (i.e. permit) or disable (i.e. forbid) events. We denote by $\mathrm{Sup}(\lambda, \sigma) \in \{1, 0\}$ the decision taken by *SUP* on an event σ when the plant has executed a trace $\lambda \in L$. $\mathrm{Sup}(\lambda, \sigma) = 1$ (resp. 0) means that σ is enabled (resp. disabled) after the execution of λ. A fundamental property of control is that: $\forall \lambda \in L : \sigma \in \Sigma_{uc} \Rightarrow \mathrm{Sup}(\lambda, \sigma) = 1$

Since $\mathcal{E}_\sigma$ and $\mathcal{D}_\sigma$ (Def. 2.2) are fundamental in the formulation of multi-decision control, we will formulate several usual notions of control as function of $\mathcal{E}_\sigma$ and $\mathcal{D}_\sigma$. First of all, the objective of *SUP* to force the plant to respect the specification can be formulated by Eq. (1,2) for every $\sigma \in \Sigma_c$. To be precise, we must say that the objective of *SUP* is to satisfy Eq. (1,2) w.r.t $(\mathcal{E}_\sigma, \mathcal{D}_\sigma)$, for every $\sigma \in \Sigma_c$. Note that these equations do not specify a decision when $\lambda \notin (\mathcal{E}_\sigma \cup \mathcal{D}_\sigma)$. This is because σ is accepted by the plant only in $\mathcal{E}_\sigma \cup \mathcal{D}_\sigma$, and hence enablement/disablement of σ has no effect on the plant when $\lambda \notin (\mathcal{E}_\sigma \cup \mathcal{D}_\sigma)$.

$$\lambda \in \mathcal{E}_\sigma \Rightarrow \mathrm{Sup}(\lambda, \sigma) = 1 \tag{1}$$

$$\lambda \in \mathcal{D}_\sigma \Rightarrow \mathrm{Sup}(\lambda, \sigma) = 0 \tag{2}$$

We have also the fundamental notion of controllable specification which can be formulated by:

Definition 2.3. *K is said L-controllable if $\forall \sigma \in \Sigma_{uc}$: $\mathcal{D}_\sigma = \emptyset$.*

The following Def. 2.4 is convenient for the formulation of multi-decision control that is presented in Section 3.

Definition 2.4. *A pair of languages (E, D) is called the enabling-pair of a control architecture $\mathcal{S}$ for $\sigma \in \Sigma_c$ to mean that they are the greatest sublanguages of L for which $\mathcal{S}$ satisfies Eq. (1,2) w.r.t (E, D).*

2.2 Decentralized Supervisory Control

Decentralized control consists in using n local supervisors $(SUP_i)_{1 \le i \le n}$, where each SUP_i has its own set of observable events $\Sigma_{o,i}$ and own set of controllable events $\Sigma_{c,i}$. We define $\Sigma_o = \Sigma_{o,1} \cup \cdots \cup \Sigma_{o,n}$ and $\Sigma_c = \Sigma_{c,1} \cup \cdots \cup \Sigma_{c,n}$. We denote by $I = \{1, \cdots, n\}$ the indexing set of all local supervisors, and by $I_\sigma = \{i \in I \,|\, \sigma \in \Sigma_{c,i}\}$ the indexing set of local supervisors controlling $\sigma \in \Sigma_c$. Let n_σ denote the cardinality of I_σ.

Among the most known decentralized control architectures developed in the literature, in the present study we consider C&P∨D&A control (Yoo and Lafortune, 2002) and inference-based control (Kumar and Takai, 2007). We consider C&P∨D&A control

because it regroups the C&P and D&A control architectures, which are the simplest relevant decentralized control architectures. We consider inference-based control because, to our best knowledge, it is the most general decentralized control before the development of multi-decision control. Interestingly, the authors of (Kumar and Takai, 2007) prove that C&P∨D&A control is a particular case of inference-based control (see Sect. 2.4).

To every control architecture is associated a property of observability, which is usually termed coobservability in decentralized control. Coobservability is fundamental because it is a necessary condition for the existence of a supervisor that satisfies Eq. (1,2). More precisely, it is shown in the literature related to every developed decentralized control, that the existence of supervisor satisfying Eq. (1,2) has as *necessary and sufficient* condition the conjunction of three properties: L_m-closure (Def. 2.1), L-controllability (Def. 2.3), and coobservability. L-controllability and L_m-closure are classical notions that are independent of the control architecture, and hence are not relevant to compare control architectures. Therefore, without loss of generality, we will consider uniquely coobservability as condition of existence of supervisor.

The following subsections 2.3 and 2.4 present succinctly C&P∨D&A control and inference-based control. In fact, we present only the notions that are indispensable to make our paper self-contained. See (Yoo and Lafortune, 2002; Kumar and Takai, 2007) for detailed studies of these two architectures.

2.3 C&P∨D&A Control

Since C&P∨D&A control is the combination of C&P control and D&A control, let us first present each of these two control architectures.

In C&P control (C for Conjunctive and P for Permissive):

1. Each local supervisor SUP_i is *permissive*, in the sense that it locally disables $\sigma \in \Sigma_{c,i}$ if and only if it is certain that $\sigma \notin \mathcal{E}_\sigma$. Formally, for any $\sigma \in \Sigma_{c,i}$, after the execution of $\lambda \in L$, SUP_i observes $P_i(\lambda)$ and computes its local decision $\mathrm{Sup}_i(P_i(\lambda),\sigma)$ by:

$$\mathrm{Sup}_i(P_i(\lambda),\sigma) = \left\{ \begin{array}{ll} 1, & \text{if } P_i(\lambda) \in P_i(\mathcal{E}_\sigma) \\ 0, & \text{if } P_i(\lambda) \notin P_i(\mathcal{E}_\sigma) \end{array} \right\} \quad (3)$$

2. The local decisions $(\mathrm{Sup}_i(P_i(\lambda),\sigma))_{i\in I_\sigma}$ are fused *conjunctively*, in order to generate the actual decision $\mathrm{Sup}(\lambda,\sigma)$. Formally:

$$\mathrm{Sup}(\lambda,\sigma) = \bigwedge_{i\in I_\sigma} \mathrm{Sup}_i(P_i(\lambda),\sigma) \quad (4)$$

The property of coobservability associated to C&P control for some $\sigma \in \Sigma_c$ is given by $\mathcal{D}_\sigma[1] = \emptyset$ where $\mathcal{D}_\sigma[1]$ is defined by Eq. (5). For any $\sigma \in \Sigma_c$, we say that $(\mathcal{E}_\sigma, \mathcal{D}_\sigma)$ is C&P-COOBS if $\mathcal{D}_\sigma[1] = \emptyset$. We have that "$(\mathcal{E}_\sigma, \mathcal{D}_\sigma)$ is C&P-COOBS" is a necessary condition so that Eqs. (3,4) guarantee the satisfaction of Eqs. (1,2) w.r.t $(\mathcal{E}_\sigma, \mathcal{D}_\sigma)$.

$$\mathcal{D}_\sigma[1] = \bigcap_{i\in I_\sigma} P_i^{-1} P_i(\mathcal{E}_\sigma) \cap \mathcal{D}_\sigma \quad (5)$$

In D&A control (D for Disjunctive and A for Anti-permissive):

1. Each local supervisor SUP_i is *anti-permissive*, in the sense that it locally enables $\sigma \in \Sigma_{c,i}$ if and only if it is certain that $\sigma \notin \mathcal{D}_\sigma$. Formally, for any $\sigma \in \Sigma_{c,i}$, after the execution of $\lambda \in L$, SUP_i observes $P_i(\lambda)$ and computes its local decision $\mathrm{Sup}_i(P_i(\lambda),\sigma)$ by:

$$\mathrm{Sup}_i(P_i(\lambda),\sigma) = \left\{ \begin{array}{ll} 0, & \text{if } P_i(\lambda) \in P_i(\mathcal{D}_\sigma) \\ 1, & \text{if } P_i(\lambda) \notin P_i(\mathcal{D}_\sigma) \end{array} \right\} \quad (6)$$

2. The local decisions $(\mathrm{Sup}_i(P_i(\lambda),\sigma))_{i\in I_\sigma}$ are fused *disjunctively*, in order to generate the actual decision $\mathrm{Sup}(\lambda,\sigma)$. Formally:

$$\mathrm{Sup}(\lambda,\sigma) = \bigvee_{i\in I_\sigma} \mathrm{Sup}_i(P_i(\lambda),\sigma) \quad (7)$$

The property of coobservability associated to D&A control for some $\sigma \in \Sigma_c$ is given by $\mathcal{E}_\sigma[1] = \emptyset$ where $\mathcal{E}_\sigma[1]$ is defined by Eq. (8). For any $\sigma \in \Sigma_c$, we say that $(\mathcal{E}_\sigma, \mathcal{D}_\sigma)$ is D&A-COOBS if $\mathcal{E}_\sigma[1] = \emptyset$. We have that "$(\mathcal{E}_\sigma, \mathcal{D}_\sigma)$ is D&A-COOBS" is a necessary condition so that Eqs. (6,7) guarantee the satisfaction of Eqs. (1,2) w.r.t $(\mathcal{E}_\sigma, \mathcal{D}_\sigma)$.

$$\mathcal{E}_\sigma[1] = \bigcap_{i\in I_\sigma} P_i^{-1} P_i(\mathcal{D}_\sigma) \cap \mathcal{E}_\sigma \quad (8)$$

We are in the presence of a C&P∨D&A control if Σ_c is partitioned in two alphabets Σ_c^{CP} and Σ_c^{DA}, such that C&P control is applied to every $\sigma \in \Sigma_c^{\mathrm{CP}}$ and D&A control is applied to every $\sigma \in \Sigma_c^{\mathrm{DA}}$. Therefore, the property of coobservability associated to C&P∨D&A control for some $\sigma \in \Sigma_c$ is that $(\mathcal{E}_\sigma, \mathcal{D}_\sigma)$ is (C&P∨D&A)-COOBS, which means $(\mathcal{E}_\sigma, \mathcal{D}_\sigma)$ is C&P-COOBS or D&A-COOBS.

2.4 Inference-Based Control

In Inference-based control, every local supervisor SUP_i generates a local decision associated to an ambiguity level which is computed in a quite complex (but systematic) way. The taken global decision is the local decision with the lowest ambiguity level. Inference-based control is based on the following iterative computations:

- Basis: $\mathcal{E}_\sigma[0] = \mathcal{E}_\sigma$ and $\mathcal{D}_\sigma[0] = \mathcal{D}_\sigma$,
- Inductive step: for $k \geq 0$

$$\mathcal{E}_\sigma[k+1] = [\bigcap_{i=1\cdots n} P_i^{-1}P_i(\mathcal{D}_\sigma[k])] \cap \mathcal{E}_\sigma[k]$$

$$\mathcal{D}_\sigma[k+1] = [\bigcap_{i=1\cdots n} P_i^{-1}P_i(\mathcal{E}_\sigma[k])] \cap \mathcal{D}_\sigma[k]$$

Inference-based control is denoted Inf_N-control if N is the maximum used ambiguity level. The property of coobservability associated to Inf_N-control for some $\sigma \in \Sigma_c$ is that $\mathcal{E}_\sigma[N+1] = \emptyset$ or $\mathcal{D}_\sigma[N+1] = \emptyset$. In such a case, we say that $(\mathcal{E}_\sigma, \mathcal{D}_\sigma)$ is Inf_N-COOBS. It is worth noting that (C&P$\vee$D&A)-COOBS is equivalent to Inf_0-COOBS.

3 MULTI-DECISION CONTROL

As we have recalled it, for every control architecture, the existence of supervisor is conditioned formally by the satisfaction of three properties: L-controllability, L_m-closure, and coobservability. The first two properties are independent of the architecture, they depend uniquely on the plant and specification models. Therefore, they are not relevant to compare control architectures. On the contrary, coobservability differs for each control architecture, and hence is a good criterion to compare control architectures. A control architecture $\mathcal{A}$ is said more general than an architecture $\mathcal{B}$ if coobservability associated to $\mathcal{A}$ is weaker than coobservability associated to $\mathcal{B}$, in the sense that the latter implies the former. Intuitively, the generality of $\mathcal{A}$ over $\mathcal{B}$ means that every language achievable by $\mathcal{B}$ is also achievable by $\mathcal{A}$.

For example inference-based control is more general than C&P$\vee$D&A control, because Inf_N-COOBS is weaker than (C&P$\vee$D&A)-COOBS. Indeed $\mathcal{E}_\sigma[1] = \emptyset$ or $\mathcal{D}_\sigma[1] = \emptyset$ implies $\mathcal{E}_\sigma[N+1] = \emptyset$ and $\mathcal{D}_\sigma[N+1] = \emptyset$ for every $N \geq 0$.

In fact, we think that one of the main motivations to research in decentralized control has been the desire to discover more and more general architectures. This is indeed the motivation of the development of the multi-decision control. The intuition of the authors of (Chakib and Khoumsi, 2011) was that a more general architecture can be obtained by using several decentralized architectures in parallel whose respective decisions are all combined to generate the effective decisions. Two combination operators have been used: disjunction and conjunction, which we present in the following subsections.

3.1 Disjunctive Multi-Decision Control

Consider several (say p) decentralized control architectures $(\mathcal{S}^j)_{j=1\cdots p}$ such that, for every $\sigma \in \Sigma_c$, the global decisions of all $\mathcal{S}^j$ are combined disjunctively to issue the effective decision on σ. Formally, we have Eq. (9), where $\mathrm{Sup}^j(\lambda,\sigma)$ is the global decision taken by $\mathcal{S}^j$ and $\mathrm{Sup}(\lambda,\sigma)$ is the effective decision synthesized from all $(\mathrm{Sup}^j(\lambda,\sigma))_{j=1\cdots p}$. The obtained architecture is named $\vee$-$(\mathcal{S}^1,\cdots,\mathcal{S}^p)$.

$$\mathrm{Sup}(\lambda,\sigma) = \bigvee_{j=1\cdots p} \mathrm{Sup}^j(\lambda,\sigma) \qquad (9)$$

Let $(\mathcal{E}^j_\sigma, \mathcal{D}^j_\sigma)$ be the enabling-pair of $\mathcal{S}^j$ for any $\sigma \in \Sigma_c$, i.e. $\mathcal{S}^j$ enables σ in $\mathcal{E}^j_\sigma$ and disables it in $\mathcal{D}^j_\sigma$. From Eq. (9), it is easy to deduce that the architecture resulting from the disjunctive combination of $(\mathcal{S}^j)_{j=1\cdots p}$ has its enabling-pair $(\mathcal{E}_\sigma, \mathcal{D}_\sigma)$ for $\sigma \in \Sigma_c$ specified by: $\mathcal{E}_\sigma = \bigcup_{j=1\cdots p} \mathcal{E}^j_\sigma$ and $\mathcal{D}_\sigma = \bigcap_{j=1\cdots p} \mathcal{D}^j_\sigma$. If we take $\mathcal{D}^j_\sigma = \mathcal{D}_\sigma$ for every $j = 1\cdots p$, we obtain that each $\mathcal{S}^j$ has an enabling-pair $(\mathcal{E}^j_\sigma, \mathcal{D}_\sigma)$ such that $(\bigcup_{j=1\cdots p} \mathcal{E}^j_\sigma, \mathcal{D}_\sigma)$ is the enabling-pair of the resulting architecture.

Consider now the opposite situation where we have to find a set of p architectures $(\mathcal{S}^j)_{j=1\cdots p}$ such that $\vee$-$(\mathcal{S}^1,\cdots,\mathcal{S}^p)$ has a given enabling pair $(\mathcal{E}_\sigma, \mathcal{D}_\sigma)$ for every $\sigma \in \Sigma_c$. This amounts to find, for every $\sigma \in \Sigma_c$, a decomposition $(\mathcal{E}^j_\sigma)_{j=1\cdots p}$ of $\mathcal{E}_\sigma$ such that every $(\mathcal{E}^j_\sigma, \mathcal{D}_\sigma)$ is the enabling-pair of $\mathcal{S}^j$.

Finding a decomposition of an infinite $\mathcal{E}_\sigma$ is in general a difficult problem if not undecidable. The authors of (Chakib and Khoumsi, 2011) have proposed that instead of decomposing $\mathcal{E}_\sigma$, we decompose the set of marked states of an FSA $\mathcal{A}_{\mathcal{E}_\sigma}$ accepting $\mathcal{E}_\sigma$. Hence, the undecidable problem of decomposing an infinite set $\mathcal{E}_\sigma$ is transformed into a decidable problem of decomposing the *finite* set of marked states of some FSA accepting $\mathcal{E}_\sigma$. With this approach, the possible decompositions are said authorized by $\mathcal{A}_{\mathcal{E}_\sigma}$ and have the characteristic that each $\mathcal{E}^j_\sigma$ corresponds to one or several marked states of $\mathcal{A}_{\mathcal{E}_\sigma}$. Note that if some marked states of $\mathcal{A}_{\mathcal{E}_\sigma}$ are split into several equivalent states, more decompositions are authorized by the new obtained FSA than by $\mathcal{A}_{\mathcal{E}_\sigma}$.

3.2 Conjunctive Multi-Decision Control

There exist strong similarities between disjunctive and conjunctive multi-decision controls. Indeed, the fundamental difference when passing from the first one to the second one is that Eq. (9) is replaced by Eq. (10), and decomposing $\mathcal{E}_\sigma$ w.r.t $\mathcal{A}_{\mathcal{E}_\sigma}$ is replaced

by decomposing $\mathcal{D}_\sigma$ w.r.t an FSA $\mathcal{A}_{\mathcal{D}_\sigma}$ accepting $\mathcal{D}_\sigma$. Hence, we obtain that each $\mathcal{S}^j$ has an enabling-pair $(\mathcal{E}_\sigma, \mathcal{D}_\sigma^j)$ such that $(\mathcal{E}_\sigma, \bigcup_{j=1\cdots p} \mathcal{D}_\sigma^j)$ is the enabling-pair of the resulting architecture. The obtained architecture is named $\wedge$-$(\mathcal{S}^1, \cdots, \mathcal{S}^p)$.

$$\mathrm{Sup}(\lambda,\sigma) = \bigwedge_{j=1\cdots p} \mathrm{Sup}^j(\lambda,\sigma) \qquad (10)$$

4 OUR PROPOSITION

We consider a plant modeled by an FSA $\mathcal{G}$ whose generated and marked languages are L and L_m, and a specification modeled by a trim FSA $\mathcal{K}$ whose marked language is K (the generated language of $\mathcal{K}$ is $\overline{K}$). We are also given the number n of sites and their respective controllable and observable alphabets $(\Sigma_{c,i})_{1\le i\le n}$ and $(\Sigma_{o,i})_{1\le i\le n}$. As explained in Sections 2.2 and 3, we consider uniquely the property of coobservability as condition of existence of supervisor.

As mentioned in the introduction, our motivation is to avoid using *complex* (namely, inference-based Inf_N) architectures by using instead several *simple* (namely, C&P$\vee$D&A) architectures running in parallel. Recall that C&P$\vee$D&A is equivalent to Inf_0, that is, it corresponds the simplest case of inference-based control, where ambiguity levels are restricted to 0. The price to be paid for this simplification is an increase of the number of architectures in parallel. For example, it is possible that a given control objective can be realized by any of the following three architectures:

- *one* Inf_2 architecture;
- *two* architectures, a Inf_0 and a Inf_1, running in parallel;
- *three* Inf_0 (i.e. C&P$\vee$D&A) architectures in parallel.

Our objective is therefore to satisfy Eqs (1,2) by using uniquely Inf_0 (i.e. C&P$\vee$D&A) control architectures in parallel. For the sake of clarity, we identify them as $Inf_0^1, Inf_0^2, \cdots$. Hence, our study consists in determining whether there exists a $\vee$-$(Inf_0^1, \cdots, Inf_0^p)$ or $\wedge$-$(Inf_0^1, \cdots, Inf_0^p)$ architecture that respects the objective formulated by Eqs. (1,2). But we present here uniquely the case of $\vee$-$(Inf_0^1, \cdots, Inf_0^p)$, because of the strong similarities between the two architectures as explained in Section 3.2. The architecture $\vee$-$(Inf_0^1, \cdots, Inf_0^p)$ is also denoted $\vee$-Inf_0^p where p specifies the number of architectures in parallel. We may also use the notation $\vee$-$Inf_0^{\geq 1}$ when p is unspecified.

To simplify the presentation of our procedure, we consider a single controllable event σ. In the presence of several controllable events, we must apply the same procedure to each of them.

4.1 Running Example

We consider the example of Figure 1 that will be used to illustrate each step of our proposition. The complete automaton represents the plant $\mathcal{G}$, and the specification $\mathcal{K}$ is obtained by removing the two dashed self-loops of σ. All states are marked, hence $\mathcal{G}$ and $\mathcal{K}$ are prefix-closed. We have two sites (i.e. $n = 2$), and the local observable and controllable alphabets are: $\Sigma_{o,1} = \{a_1, b_1, c_1\}$, $\Sigma_{o,2} = \{a_2, b_2, c_2, d_2\}$, and $\Sigma_{c,1} = \Sigma_{c,2} = \{\sigma\}$.

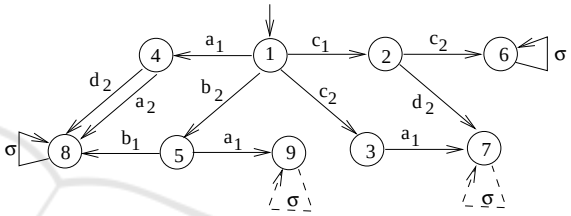

Figure 1: Plant $\mathcal{G}$ and Specification $\mathcal{K}$.

4.2 Step 1: Computing $\mathcal{A}_{\mathcal{E}_\sigma}$ and $\mathcal{A}_{\mathcal{D}_\sigma}$

Recall that $\mathcal{A}_X$ denotes an automaton accepting a language X. Since our control objective is based on $\mathcal{E}_\sigma$ and $\mathcal{D}_\sigma$ (through Eqs (1,2)), we compute $\mathcal{A}_{\mathcal{E}_\sigma}$ and $\mathcal{A}_{\mathcal{D}_\sigma}$. $\mathcal{A}_{\mathcal{E}_\sigma}$ is computed from $\mathcal{K}$ by marking uniquely the states where σ is accepted and then removing the states from which no marked state is reachable. $\mathcal{A}_{\mathcal{D}_\sigma}$ is computed in 3 steps: 1) we compute the synchronized product of $\mathcal{G}$ and $\mathcal{K}$, which is an automaton whose states are defined by (q,r), where q and r are states of $\mathcal{G}$ and $\mathcal{K}$, respectively; 2) we mark every state (q,r) where we have σ enabled in the state q of $\mathcal{G}$ and disabled in the state r of $\mathcal{K}$; and 3) we remove the states from which no marked state is reachable. Let X and Y denote the sets of states of $\mathcal{A}_{\mathcal{E}_\sigma}$ and $\mathcal{A}_{\mathcal{D}_\sigma}$, respectively. Let X_m and Y_m denote the sets of marked states of $\mathcal{A}_{\mathcal{E}_\sigma}$ and $\mathcal{A}_{\mathcal{D}_\sigma}$, respectively. The states of X_m are identified as $x^1, x^2, \cdots$, and the states of Y_m are identified as $y^1, y^2, \cdots$.

Consider our example: $\mathcal{A}_{\mathcal{E}_\sigma}$ is obtained from Figure 1 by removing states 3, 7 and 9, and marking states 6 and 8. $\mathcal{A}_{\mathcal{D}_\sigma}$ is obtained from Figure 1 by removing states 4, 6 and 8, and by marking states 7 and 9. $\mathcal{A}_{\mathcal{E}_\sigma}$ and $\mathcal{A}_{\mathcal{D}_\sigma}$ are represented in Figures 2 and 3. The marked states of $\mathcal{A}_{\mathcal{E}_\sigma}$ are x^1 and x^2, and the marked states of $\mathcal{A}_{\mathcal{D}_\sigma}$ are y^1 and y^2.

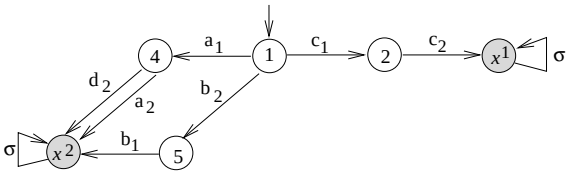

Figure 2: Automaton $\mathcal{A}_{\mathcal{E}_\sigma}$.

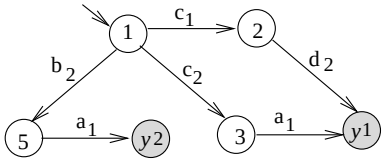

Figure 3: Automaton $\mathcal{A}_{\mathcal{D}_\sigma}$.

4.3 Step 2: Computing $\mathcal{A}_{\mathcal{E}_\sigma[1]}$ and $\mathcal{A}_{\mathcal{D}_\sigma[1]}$

Inf_0-COOBS is based on $\mathcal{D}_\sigma[1]$ and $\mathcal{E}_\sigma[1]$ defined by Eqs. (5,8). Hence, FSA $\mathcal{A}_{\mathcal{E}_\sigma[1]}$ and $\mathcal{A}_{\mathcal{D}_\sigma[1]}$ accepting $\mathcal{E}_\sigma[1]$ and $\mathcal{D}_\sigma[1]$ are computed from $\mathcal{A}_{\mathcal{E}_\sigma}$ and $\mathcal{A}_{\mathcal{D}_\sigma}$ by Eqs. (11,12), where $\bigotimes$ and $\times$ denote the synchronized product of automata.

$$\mathcal{A}_{\mathcal{E}_\sigma[1]} = \bigotimes_{i\in I_\sigma} P_i^{-1}P_i(\mathcal{A}_{\mathcal{D}_\sigma}) \times \mathcal{A}_{\mathcal{E}_\sigma} \qquad (11)$$

$$\mathcal{A}_{\mathcal{D}_\sigma[1]} = \bigotimes_{i\in I_\sigma} P_i^{-1}P_i(\mathcal{A}_{\mathcal{E}_\sigma}) \times \mathcal{A}_{\mathcal{D}_\sigma} \qquad (12)$$

1. Every state of $\mathcal{A}_{\mathcal{E}_\sigma[1]}$ is defined by $u = (u_1,\cdots,u_{n_\sigma},u_{n_\sigma+1})$, where $u_{n_\sigma+1} \in X$ and $u_i \subseteq Y$ for every $i \in I_\sigma$.
   u is said marked if $u_{n_\sigma+1} \in X_m$ and $u_i \cap Y_m \neq \emptyset$ for every $i \in I_\sigma$.
   u is said x^j-marked if it is marked and $u_{n_\sigma+1} = x^j$.
2. Every state of $\mathcal{A}_{\mathcal{D}_\sigma[1]}$ is defined by $v = (v_1,\cdots,v_{n_\sigma},v_{n_\sigma+1})$, where $v_{n_\sigma+1} \in Y$ and $v_i \subseteq X$ for every $i \in I_\sigma$.
   v is said marked if $v_{n_\sigma+1} \in Y_m$ and $v_i \cap X_m \neq \emptyset$ for every $i \in I_\sigma$.
   v is said x^j-marked if it is marked and $x^j \in v_i$ for every $i \in I_\sigma$.

Then, we remove from $\mathcal{A}_{\mathcal{E}_\sigma[1]}$ and $\mathcal{A}_{\mathcal{D}_\sigma[1]}$ the states from which no marked state is reachable.

Consider our example: $\mathcal{A}_{\mathcal{E}_\sigma[1]}$ and $\mathcal{A}_{\mathcal{D}_\sigma[1]}$ are represented in Figures 4 and 5, where marked states are dark and x^j-marked states are indicated by x^j.

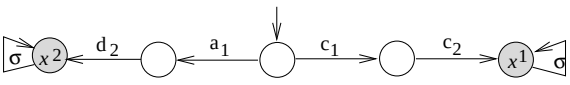

Figure 4: Automaton $\mathcal{A}_{\mathcal{E}_\sigma[1]}$.

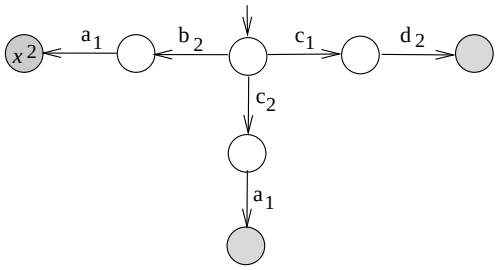

Figure 5: Automaton $\mathcal{A}_{\mathcal{D}_\sigma[1]}$.

4.4 Step 3: Checking if $(\mathcal{E}_\sigma,\mathcal{D}_\sigma)$ is $\vee$-$Inf_0^{\geq 1}$-COOBS w.r.t $\mathcal{A}_{\mathcal{E}_\sigma}$

First of all, it is worth checking whether Inf_0-architecture (without multi-decision) can be used to satisfy Eqs. (1,2) w.r.t $(\mathcal{E}_\sigma,\mathcal{D}_\sigma)$. As explained in Section 2.3, this amounts to determine if $(\mathcal{E}_\sigma,\mathcal{D}_\sigma)$ is Inf_0-COOBS. The verification is done as follows: $(\mathcal{E}_\sigma,\mathcal{D}_\sigma)$ is Inf_0-COOBS if and only if $\mathcal{A}_{\mathcal{E}_\sigma[1]}$ or $\mathcal{A}_{\mathcal{D}_\sigma[1]}$ has no marked state.

If we compute that $(\mathcal{E}_\sigma,\mathcal{D}_\sigma)$ is Inf_0-COOBS, Inf_0-architecture can be used since it satisfies Eqs. (1,2) w.r.t $(\mathcal{E}_\sigma,\mathcal{D}_\sigma)$; we go to Step 5 (Sect. 4.6) to compute the local and global decisions taken by the architecture.

If we compute that $(\mathcal{E}_\sigma,\mathcal{D}_\sigma)$ is not Inf_0-COOBS (i.e. both $\mathcal{A}_{\mathcal{E}_\sigma[1]}$ or $\mathcal{A}_{\mathcal{D}_\sigma[1]}$ have marked states), we continue in this step 3.

Since $(\mathcal{E}_\sigma,\mathcal{D}_\sigma)$ is not Inf_0-COOBS, we have now to verify if multi-decision can help. More precisely, we have to determine whether there exists a $\vee$-$Inf_0^{\geq 1}$ architecture that satisfies Eqs. (1,2) w.r.t $(\mathcal{E}_\sigma,\mathcal{D}_\sigma)$. As explained in Section 3.1, this amounts to determine whether there exists a decomposition $(\mathcal{E}_\sigma^j)_{j=1\cdots}$ of $\mathcal{E}_\sigma$ such that $(\mathcal{E}_\sigma^j,\mathcal{D}_\sigma)_{j=1\cdots}$ are the respective enabling-pairs of the Inf_0 architectures in parallel. From Section 2.3, this amounts to determine whether there exists a decomposition $(\mathcal{E}_\sigma^j)_{j=1\cdots}$ of $\mathcal{E}_\sigma$ such that every $(\mathcal{E}_\sigma^j,\mathcal{D}_\sigma)$ is Inf_0-COOBS, i.e. $\mathcal{E}_\sigma^j[1]=\emptyset$ or $\mathcal{D}_\sigma^j[1]=\emptyset$ for every $j = 1\cdots p$, where $\mathcal{D}_\sigma^j[1]$ and $\mathcal{E}_\sigma^j[1]$ are defined as $\mathcal{D}_\sigma[1]$ and $\mathcal{E}_\sigma[1]$ in Eqs. (5,8), but by using $\mathcal{E}_\sigma^j$ instead of $\mathcal{E}_\sigma$.

We have explained in the last paragraph of Section 3.1 why we consider uniquely decompositions $(\mathcal{E}_\sigma^j)_{j=1\cdots}$ authorized by $\mathcal{A}_{\mathcal{E}_\sigma}$, i.e. where each $\mathcal{E}_\sigma^j$ corresponds to one or several marked states of $\mathcal{A}_{\mathcal{E}_\sigma}$. Therefore, by using Def. 4.1 below, our objective becomes to determine if $(\mathcal{E}_\sigma,\mathcal{D}_\sigma)$ is $\vee$-$Inf_0^{\geq 1}$-COOBS w.r.t $\mathcal{A}_{\mathcal{E}_\sigma}$.

Definition 4.1. *$(\mathcal{E}_\sigma,\mathcal{D}_\sigma)$ is said $\vee$-$Inf_0^{\geq 1}$-COOBS w.r.t $\mathcal{A}_{\mathcal{E}_\sigma}$ if there exists a decomposition $(\mathcal{E}_\sigma^j)_{j=1\cdots}$*

of $\mathcal{E}_\sigma$ which is authorized by $\mathcal{A}_{\mathcal{E}_\sigma}$, such that every $(\mathcal{E}_\sigma^j, \mathcal{D}_\sigma)$ is Inf_0-COOBS.

Let us consider the decomposition (in fact a partition) of $\mathcal{E}_\sigma$ where each $\mathcal{E}_\sigma^j$ corresponds to a single marked state $x^j \in X_m$ of $\mathcal{A}_{\mathcal{E}_\sigma}$. Such a partition is called trivial partition of $\mathcal{E}_\sigma$ w.r.t $\mathcal{A}_{\mathcal{E}_\sigma}$. Note that the number p of languages $\mathcal{E}_\sigma^j$ of such a trivial partition is the cardinality of the set X_m of marked states of $\mathcal{A}_{\mathcal{E}_\sigma}$. Clearly, if for such a trivial partition we have not that every $(\mathcal{E}_\sigma^j, \mathcal{D}_\sigma)$ is Inf_0-COOBS, then $(\mathcal{E}_\sigma, \mathcal{D}_\sigma)$ is not $\vee$-$Inf_0^{\geq 1}$-COOBS w.r.t $\mathcal{A}_{\mathcal{E}_\sigma}$, i.e. there exists no other decomposition authorized by $\mathcal{A}_{\mathcal{E}_\sigma}$ for which we have that every $(\mathcal{E}_\sigma^j, \mathcal{D}_\sigma)$ is Inf_0-COOBS. Therefore, $(\mathcal{E}_\sigma, \mathcal{D}_\sigma)$ is $\vee$-$Inf_0^{\geq 1}$-COOBS w.r.t $\mathcal{A}_{\mathcal{E}_\sigma}$ if and only if for every $\mathcal{E}_\sigma^j$ corresponding to a state $x^j \in X_m$ of $\mathcal{A}_{\mathcal{E}_\sigma}$, $(\mathcal{E}_\sigma^j, \mathcal{D}_\sigma)$ is Inf_0-COOBS, i.e. $\mathcal{E}_\sigma^j[1] = \emptyset$ or $\mathcal{D}_\sigma^j[1] = \emptyset$. This can be verified as follows: Emptiness of $\mathcal{E}_\sigma^j[1]$ (resp. $\mathcal{D}_\sigma^j[1]$) is equivalent to that $\mathcal{A}_{\mathcal{E}_\sigma[1]}$ (resp. $\mathcal{A}_{\mathcal{D}_\sigma[1]}$) has no x^j-marked state.

If we compute that $(\mathcal{E}_\sigma, \mathcal{D}_\sigma)$ is $\vee$-$Inf_0^{\geq 1}$-COOBS w.r.t $\mathcal{A}_{\mathcal{E}_\sigma}$, the $\vee$-$Inf_0^{\geq 1}$-architecture can be used since it satisfies Eqs. (1,2) w.r.t $(\mathcal{E}_\sigma, \mathcal{D}_\sigma)$. We go to Step 5 (Sect. 4.6) to compute the decisions taken by the architecture.

If we compute that $(\mathcal{E}_\sigma, \mathcal{D}_\sigma)$ is not $\vee$-$Inf_0^{\geq 1}$-COOBS w.r.t $\mathcal{A}_{\mathcal{E}_\sigma}$, we go to step 4 (Sect. 4.5).

Consider our example: Both $\mathcal{A}_{\mathcal{E}_\sigma[1]}$ and $\mathcal{A}_{\mathcal{D}_\sigma[1]}$ have marked states (dark in Figs. 4 and 5), hence $(\mathcal{E}_\sigma, \mathcal{D}_\sigma)$ is not Inf_0-COOBS. We consider the trivial partition $\mathcal{E}_\sigma^1$ and $\mathcal{E}_\sigma^2$ corresponding to states x^1 and x^2 of $\mathcal{A}_{\mathcal{E}_\sigma}$ of Fig. 2. The languages of $\mathcal{A}_{\mathcal{E}_\sigma}$ corresponding to x^1 and x^2 are, respectively, $\mathcal{E}_\sigma^1 = \{c_1c_2\sigma^*\}$ and $\mathcal{E}_\sigma^2 = \{a_1d_2\sigma^*, a_1a_2\sigma^*, b_2b_1\sigma^*\}$. x^1 is not problematic because $\mathcal{A}_{\mathcal{D}_\sigma[1]}$ of Fig. 5 has no x^1-marked state, which means that $\mathcal{D}_\sigma^1[1] = \emptyset$. Therefore, $(\mathcal{E}_\sigma^1, \mathcal{D}_\sigma)$ is Inf_0-COOBS. Hence, $\mathcal{E}_\sigma^1 = \{c_1c_2\sigma^*\}$ will be kept as it is. x^2 is problematic because both $\mathcal{A}_{\mathcal{E}_\sigma[1]}$ and $\mathcal{A}_{\mathcal{D}_\sigma[1]}$ have a x^2-marked state, which means that $\mathcal{E}_\sigma^2[1] \neq \emptyset$ and $\mathcal{D}_\sigma^2[1] \neq \emptyset$. Therefore, $(\mathcal{E}_\sigma^2, \mathcal{D}_\sigma)$ is not Inf_0-COOBS. This problem is solved in the following step (Sect. 4.5).

4.5 Step 4: When $(\mathcal{E}_\sigma, \mathcal{D}_\sigma)$ is not $\vee$-$Inf_0^{\geq 1}$-COOBS w.r.t $\mathcal{A}_{\mathcal{E}_\sigma}$

If $(\mathcal{E}_\sigma, \mathcal{D}_\sigma)$ is not $\vee$-$Inf_0^{\geq 1}$-COOBS w.r.t $\mathcal{A}_{\mathcal{E}_\sigma}$, we have to determine if we can obtain that $(\mathcal{E}_\sigma, \mathcal{D}_\sigma)$ is $\vee$-$Inf_0^{\geq 1}$-COOBS w.r.t another automaton which authorizes more decompositions than $\mathcal{A}_{\mathcal{E}_\sigma}$. In fact, we need to consider only the problematic states, i.e. the states x^j of $\mathcal{A}_{\mathcal{E}_\sigma}$ that correspond to the $\mathcal{E}_\sigma^j$ for which $\mathcal{E}_\sigma^j[1] \neq \emptyset$ and $\mathcal{D}_\sigma^j[1] \neq \emptyset$. The solution is to split each problematic state x^j into several equivalent states $x^{j_1}, \cdots, x^{j_j}$, which permits to decompose the corresponding problematic $\mathcal{E}_\sigma^j$ language into several languages $\mathcal{E}_\sigma^{j_1}, \cdots, \mathcal{E}_\sigma^{j_j}$. Hence, we need to use an operator that splits states of an automaton without changing its accepted language. Let us consider the example of operator O given by Eq. (13).

$$O(\mathcal{A}_{\mathcal{E}_\sigma}) = P_1^{-1}P_1(\mathcal{A}_{\mathcal{E}_\sigma}) \times \cdots P_n^{-1}P_n(\mathcal{A}_{\mathcal{E}_\sigma}) \times \mathcal{A}_{\mathcal{E}_\sigma} \quad (13)$$

Once $O(\mathcal{A}_{\mathcal{E}_\sigma})$ computed, we repeat Steps 2-3 (Sections 4.3 and 4.4), but by using $O(\mathcal{A}_{\mathcal{E}_\sigma})$ instead of $\mathcal{A}_{\mathcal{E}_\sigma}$ and by considering as marked states of $O(\mathcal{A}_{\mathcal{E}_\sigma})$ uniquely the states equivalent to the problematic states of $\mathcal{A}_{\mathcal{E}_\sigma}$. And so on, we may have to do the iteration steps 2-3-4-2-...-3, and hence compute $\mathcal{A}_{\mathcal{E}_{\sigma 1}}, \mathcal{A}_{\mathcal{E}_{\sigma 2}}, \cdots$. The iteration is stopped in Step 3 if we obtain that $(\mathcal{E}_\sigma, \mathcal{D}_\sigma)$ is $\vee$-$Inf_0^{\geq 1}$-COOBS w.r.t the current $\mathcal{A}_{\mathcal{E}_{\sigma k}}$, or after a given number of iterations.

Consider our example: We have seen that for the $\mathcal{E}_\sigma^2$ corresponding to state x^2 of $\mathcal{A}_{\mathcal{E}_\sigma}$ (Fig. 2), $(\mathcal{E}_\sigma^2, \mathcal{D}_\sigma)$ is not Inf_0-COOBS. If we apply the operator of Eq. (13) to $\mathcal{A}_{\mathcal{E}_\sigma}$ of Fig. 2, we obtain $O(\mathcal{A}_{\mathcal{E}_\sigma})$ of Fig. 6. By this operation, the problematic state x^2 of $\mathcal{A}_{\mathcal{E}_\sigma}$ has been split into the two states indicated by x^2 and x^3 in Fig. 6. This amounts to decompose the language $\mathcal{E}_\sigma^2 = \{a_1d_2\sigma^*, a_1a_2\sigma^*, b_2b_1\sigma^*\}$ into the two languages $\mathcal{E}_\sigma^2 = \{a_1d_2\sigma^*, a_1a_2\sigma^*\}$ and $\mathcal{E}_\sigma^3 = \{b_2b_1\sigma^*\}$.

Since the language $\mathcal{E}_\sigma^1$ corresponding to x^1 needs not be changed, we consider as marked uniquely the states x^2 and x^3. New $\mathcal{A}_{\mathcal{E}_\sigma[1]}$ and $\mathcal{A}_{\mathcal{D}_\sigma[1]}$ are computed by applying Eqs. (11,12) to $(O(\mathcal{A}_{\mathcal{E}_\sigma}), \mathcal{A}_{\mathcal{D}_\sigma})$. We find that the new $\mathcal{A}_{\mathcal{E}_\sigma[1]}$ and $\mathcal{A}_{\mathcal{D}_\sigma[1]}$ have no x^2-marked or x^3-marked automaton. This means that $(\mathcal{E}_\sigma^2, \mathcal{D}_\sigma)$ and $(\mathcal{E}_\sigma^3, \mathcal{D}_\sigma)$ are Inf_0-COOBS, To recapitulate, we have found a partition $(\mathcal{E}_\sigma^j)_{j=1,2,3}$ of $\mathcal{E}_\sigma$ such that every $(\mathcal{E}_\sigma^j, \mathcal{D}_\sigma)$ is Inf_0-COOBS, i.e. for each $j = 1,2,3$ we have $\mathcal{E}_\sigma^j[1] = \emptyset$ or $\mathcal{D}_\sigma^j[1] = \emptyset$. Indeed, we obtain $\mathcal{D}_\sigma^j[1] = \emptyset$ for $j = 1,2,3$, and $\mathcal{E}_\sigma^2[1] = \emptyset$ and $\mathcal{E}_\sigma^3[1] = \emptyset$. Therefore, $(\mathcal{E}_\sigma, \mathcal{D}_\sigma)$ is $\vee$-Inf_0^3-COOBS w.r.t $O(\mathcal{A}_{\mathcal{E}_\sigma})$.

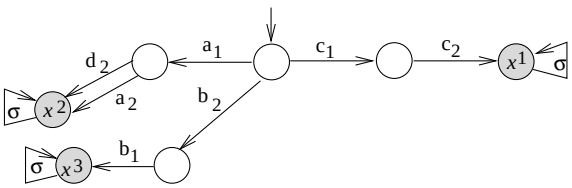

Figure 6: Automaton $O(\mathcal{A}_{\mathcal{E}_\sigma})$.

4.6 Step 5: After Steps 2-3-4-2-...2-3

If after Steps 2-3-4-2-...2-3, we obtain that $(\mathcal{E}_\sigma, \mathcal{D}_\sigma)$ is not $\vee$-$Inf_0^{\geq 1}$-COOBS w.r.t the current $\mathcal{A}_{\mathcal{E}_{\sigma k}}$, we conclude that our method is not applicable with the proposed operator O that computes $\mathcal{A}_{\mathcal{E}_{\sigma 1}}, \mathcal{A}_{\mathcal{E}_{\sigma 2}}, \cdots$. Otherwise (i.e. $(\mathcal{E}_\sigma, \mathcal{D}_\sigma)$ is $\vee$-$Inf_0^{\geq 1}$-COOBS w.r.t the current $\mathcal{A}_{\mathcal{E}_{\sigma k}}$), we consider the decomposition $(\mathcal{E}_\sigma^j)_{j=1\cdots p}$ that has been constructed. We can use the $\vee$-$(Inf_0^1, \cdots, Inf_0^p)$ architecture such that each $(\mathcal{E}_\sigma^j, \mathcal{D}_\sigma)$ is the enabling-pair of the Inf_0^j-architecture. The effective decision $\mathrm{Sup}(\lambda, \sigma)$ is computed by Eq. (9) where each $\mathrm{Sup}^j(\lambda, \sigma)$ is the decision taken by Inf_0^j. Each $\mathrm{Sup}^j(\lambda, \sigma)$ is computed as follows:

- If $\mathcal{D}_\sigma^j[1] = \emptyset$: Therefore, Inf_0^j is a C&P-architecture and hence its global decision $\mathrm{Sup}^j(\lambda, \sigma)$ is computed by Eqs. (3,4), but by adding a superscript j to $\mathcal{E}_\sigma$, $\mathrm{Sup}_i(P_i(\lambda), \sigma)$ and $\mathrm{Sup}(\lambda, \sigma)$.
- If $\mathcal{E}_\sigma^j[1] = \emptyset$: Therefore, Inf_0^j is a D&A-architecture and hence its global decision $\mathrm{Sup}^j(\lambda, \sigma)$ is computed by Eqs. (6,7), but by adding a superscript j to $\mathrm{Sup}_i(P_i(\lambda), \sigma)$ and $\mathrm{Sup}(\lambda, \sigma)$.

Note that the case where $(\mathcal{E}_\sigma, \mathcal{D}_\sigma)$ is Inf_0-COOBS can be treated as above by taking $p = 1$.

Consider our example: we have found a partition $(\mathcal{E}_\sigma^j)_{j=1,2,3}$ of $\mathcal{E}_\sigma$ such that every $(\mathcal{E}_\sigma^j, \mathcal{D}_\sigma)$ is C&P-COOBS. Recall that $\mathcal{E}_\sigma^1 = \{c_1c_2\sigma^*\}$, $\mathcal{E}_\sigma^2 = \{a_1d_2\sigma^*, a_1a_2\sigma^*\}$ and $\mathcal{E}_\sigma^3 = \{b_2b_1\sigma^*\}$. Therefore, we can use three C&P-architectures in parallel. Table 1 represents the generated decisions for the traces $\lambda \in \mathcal{E}_\sigma \cup \mathcal{D}_\sigma$, which are the only traces where a control decision on σ is relevant. Columns P_1 and P_2 contain the projections $P_1(\lambda)$ and $P_2(\lambda)$. For each $j = 1, 2, 3$, we have three columns S_1^j, S_2^j, S^j that contain the decisions of the j^{th} architecture: S_1^j and S_2^j are the local decisions computed by Eq. (3) for each of the two sites; S^j is the global decision computed by Eq. (4), i.e. S^j is the conjunction of S_1^j and S_2^j. The last column contains the effective decision computed Eq. (9), i.e. S is the disjunction of S^1, S^2, S^3. We see that the decision 1 is taken for the traces of $\mathcal{E}_\sigma$ (lines 1-4), and the decision 0 is taken for the traces of $\mathcal{D}_\sigma$ (lines 5-7).

Table 1: The decisions taken by the $\vee$-Inf_0^3 architecture of our example.

λ	P_1	P_2	S_1^1	S_2^1	S^1	S_1^2	S_2^2	S^2	S_1^3	S_2^3	S^3	S
$c_1c_2\sigma^*$	c_1	c_2	1	1	1	0	0	0	0	0	0	1
$b_2b_1\sigma^*$	b_1	b_2	0	0	0	1	1	1	0	0	0	1
$a_1d_2\sigma^*$	a_1	d_2	0	0	0	0	0	0	1	1	1	1
$a_1a_2\sigma^*$	a_1	a_2	0	0	0	0	0	0	1	1	1	1
c_1d_2	c_1	d_2	1	0	0	0	0	0	0	0	1	0
c_2a_1	a_1	c_2	0	1	0	0	0	0	1	0	0	0
b_2a_1	a_1	b_2	0	0	0	0	0	0	1	0	0	0

5 CONCLUSION

Multi-decision control has been so far studied as a solution to generalize inference-based control, by using several inference-based architectures in parallel. In the present study, we have used multi-decision control with a different perspective. Instead of using it to *generalize* inference-based control, we have rather used it as an *alternative* to inference-based control. More precisely, our objective has been to avoid using inference-based architectures, by using instead several simple architectures running in parallel. The simple used architectures are called C&P$\vee$D&A, which are the simplest relevant decentralized architectures known in the literature.

A possible future work is to develop a more general procedure that tries architectures with the smallest nonnull ambiguity levels, if the control objective cannot be reached by uniquely the ambiguity level 0.

REFERENCES

Chakib, H. and Khoumsi, A. (2011). Multi-decision Supervisory Control: Parallel Decentralized Architectures Cooperating for Controlling Discrete Event Systems. *IEEE Transactions on Automatic Control*, 56(11):2608–2622.

Kumar, R. and Takai, S. (2007). Inference-Based Ambiguity Management in Decentralized Decision-Making: Decentralized Control of Discrete Event Systems. *IEEE Transactions on Automatic Control*, 52(10):1783–1794.

Rudie, K. and Wonham, W. (1992). Think globally, act locally: Decentralized supervisory control. *IEEE Transactions on Automatic Control*, 37(11):1692–1708.

Yoo, T.-S. and Lafortune, S. (2002). A General Architecture for Decentralized Supervisory Control of Discrete-Event Systems. *Discrete Event Dyna. Syst.: Theory Applicat.*, 12:335–377.