Automatica 185 (2026) 112773

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Decentralized control of finite state systems: A game theoretic approach☆

Giordano Pola *, Elena De Santis, Maria Domenica Di Benedetto

Department of Information Engineering, Computer Science, and Mathematics, University of L'Aquila, Via Vetoio, Coppito, 67100 L'Aquila, Italy
Center of Excellence for Research DEWS, University of L'Aquila, Via Vetoio, Coppito, 67100 L'Aquila, Italy

ARTICLE INFO

Article history:
Received 12 February 2024
Received in revised form 18 September 2025
Accepted 21 November 2025

Keywords:
Finite state systems
Decentralized control
Supervisory control
Regular languages
Games

ABSTRACT

In this paper we consider a pair of interconnected, nondeterministic and metric finite state systems and address a control problem where controllers are designed for enforcing local specifications expressed in terms of regular languages, up to a desired accuracy. The control architecture considered is decentralized, that is each controller can only communicate with the corresponding plant. Since plant systems are interconnected, the part of the specification that can be enforced on one system depends on the part that can be applied on the other one. We show how this dependency can be formalized in terms of equilibria, by extending game theory to the present framework. We introduce notions of equilibria, Nash equilibria and dominant equilibria. When controlled plants are at an equilibrium, they satisfy a part of their specification; when they are at a Nash equilibrium, deviation of each plant from its control strategy may correspond to a loss in terms of the part of specification enforced; when they are at a dominant equilibrium, there is no other equilibrium where plants can achieve larger parts of the corresponding specifications. A characterization of these notions is derived and checkable conditions are discussed. An example in the context of multi-agent systems with shared resources is also included.

1. Introduction

Decentralized control offers a methodology to controllers' synthesis that is effective in taming complexity of large-scale real-world systems. Several decentralized control techniques have been proposed in the literature, which include decentralized stabilization and regulation, decentralized robust stabilization optimization and reliability design, decentralized adaptive control, consensus and formation control problems in multi-agent systems, applications to mobile robotics, decentralized supervisory control of discrete-event systems (DES).

In this paper, we consider a system given by the interconnection of a pair of nondeterministic and metric finite state systems P_1 and P_2, and address a decentralized control problem where two controllers, C_1 for plant P_1 and C_2 for plant P_2, are designed to enforce desired local regular language specifications Q_1 and Q_2 on P_1 and P_2 respectively, up to desired accuracies. Each controller shares information only with the corresponding plant. As discussed e.g. in Pola and Di Benedetto (2019) and Tabuada (2008), regular languages are useful for modeling specifications of interest in e.g. cyber–physical systems. The control scheme considered, also depicted in Fig. 1, naturally arises for example in multi-agent systems and consensus problems. Since plants are interconnected, the part of the specification that can be enforced on one system depends on the one on the other system and hence control strategies may not be compatible. A straightforward approach to solve this control problem follows this reasoning: when designing controller C_1 to enforce specification Q_1 on plant P_1, the state of plant P_2, which is an external input for P_1, is treated as an unknown disturbance, and the same reasoning by reversing indexes 1 and 2. As pointed out in this paper through an example, this approach may be restrictive because some solutions to the control problem may exist without being captured by using this reasoning. One additional issue complicates things even more. We show through an example that more than one solution to the control problem may exist, but in general a maximum over the solutions (in terms of amount of specification enforced) does not exist.

We deal with these issues by resorting to a game theoretic approach. We interpret our control problem as a game between

☆ This work is partially supported by the Center of Excellence DEWS. The material in this paper was partially presented at The 62nd IEEE Conference on Decision and Control (CDC), December 13–15, 2023, Marina Bay Sands, Singapore. This paper was recommended for publication in revised form by Associate Editor Michel Reniers under the direction of Editor Christos G. Cassandras.

* Corresponding author at: Department of Information Engineering, Computer Science, and Mathematics, University of L'Aquila, Via Vetoio, Coppito, 67100 L'Aquila, Italy.

E-mail addresses: giordano.pola@univaq.it (G. Pola), elena.desantis@univaq.it (E. De Santis), mariadomenica.dibenedetto@univaq.it (M.D. Di Benedetto).

https://doi.org/10.1016/j.automatica.2025.112773

two players identified with the plants. Action of each player is formalized by the notion of strategy defined as a controller and a relation between initial states of the plant and of the controller. The goal of each player is to enforce a local specification on its plant. We propose a notion of equilibrium between strategies so that, at equilibrium, there exists a strategy for each player P_i, $i = 1, 2$, such that a sublanguage of the local specification is respected, given the strategy of the other player P_{3-i}, $i = 1, 2$, that is unknown to P_i, $i = 1, 2$. We then propose a stronger notion of equilibrium, where deviation of each plant from its control strategy may correspond to a loss in terms of the part of specification enforced. We call this notion Nash equilibrium because it resembles the standard notion given for discrete, continuous and hybrid processes, see Remark 1. We also propose a notion of dominant equilibrium where there is no other pair of strategies enforcing larger parts of specifications than the ones in the dominant equilibria. Sets of equilibria, Nash and dominant equilibria are either empty or with infinite cardinality, which makes the problem of their characterization challenging.

To this purpose, we propose an algorithm that takes as input, plants P_1, P_2 and a finite collection of sub-transition systems of the transition systems generating specifications Q_1 and Q_2. Output of the algorithm is a collection of pairs of strategies, solving the control problem considered, and is denoted by $\mathcal{G}$. Since the algorithm is run with a finite collection of inputs, set $\mathcal{G}$ is finite. We show that for any equilibrium, Nash equilibrium and dominant equilibrium, there exists at least a pair of strategies in $\mathcal{G}$ enforcing the same part of the specifications. Since set $\mathcal{G}$ is finite, a characterization of the sets of equilibria, Nash and dominant equilibria is a decidable problem. An example within multi-agent systems with shared resources is also included.

A comparison with related work follows. The plants' models considered in this paper can be viewed as a subclass of nondeterministic DES, because they are not equipped with marked states, while DES are. On the other hand, our models are equipped with metric in the set of states, which is not the case in the traditional formulation of DES. Decentralized control for DES was extensively studied in the literature, starting from the well-known supervisory control theory (SCT) built by Wonham and Ramadge in the 80's of the last century. We recall here Rudie and Wonham (1992) and Jiang et al. (2010). Work (Rudie & Wonham, 1992) was among the first, if not, to address decentralized control for DES. It considers decentralized control architectures with *global* specifications. The second one, instead, considers decentralized control architectures with *local* specifications, as in our framework. The key distinction between SCT and our framework lies in their settings. Here, system behavior is defined as strings of system states, whereas in SCT as strings of events. The role of controllers also differs: in SCT, controllers disable controllable events to restrict system transitions, while in this paper, controllers execute control inputs to select transitions. Moreover, considering metrics in the set of states, which naturally translates in the requirement to satisfy the specifications in an approximate sense, offers more flexibility in finding solutions to the control problem for finite state systems from an engineering perspective. Considering metrics in the set of states also makes it possible to deal with purely continuous or hybrid systems that can be approximated by nondeterministic metric finite state systems. Literature in this regard is rather rich, see e.g. Belta et al. (2017), Girard and Pappas (2007), Pola and Di Benedetto (2019) and Tabuada (2009) and the references therein. For example, for networks of discrete-time nonlinear systems with arbitrary topological interconnection, Pola et al. (2016) proposes approximations by discrete abstractions, and Pola et al. (2018) decentralized control for deterministic plants, with dynamic and open-loop controllers, and global specifications. Our recent work (Pola et al., 2024) is related to this paper and addresses decentralized control of finite and metric nondeterministic systems from two agents to an arbitrarily large number of agents. However Pola et al. (2024), as well as references cited above, do not provide a characterization of solutions in terms of Nash and dominant equilibria, which is the *contribution* of the present paper. Initial studies in this regard were presented in the conference publication (Pola et al., 2023).

This paper is organized as follows. Section 2 includes notation, preliminary and new definitions that are instrumental to derive our results. Section 3 introduces interconnected finite state systems and the control problem formulation. Section 4 introduces the notions of equilibria, Nash and dominant equilibria. Section 5 presents the results obtained and Section 6 an illustrative example. Section 7 offers some concluding remarks.

2. Notation and preliminary definitions

Symbol $\wedge$ denotes the logical conjunction and symbol $\varnothing$ the empty set. Symbols $\mathbb{N}$, $\mathbb{R}$, and $\mathbb{R}_0^+$ denote the set of non-negative integer, real, and non-negative real numbers, respectively. Given $a, b \in \mathbb{N}$ we set $[a; b] = \{x \in \mathbb{N} \mid a \leq x \leq b\}$. Symbol $|a|$ denotes the absolute value of $a \in \mathbb{R}$. The sum of sets $X_1 \subseteq \mathbb{R}^n$ and $X_2 \subseteq \mathbb{R}^n$ is the set $X_1 + X_2 := \{z \in \mathbb{R}^n \mid \exists x_1 \in X_1 \wedge \exists x_2 \in X_2 \text{ s.t. } z = x_1 + x_2\}$.

Given a finite set X, card(X) denotes the cardinality of X, 2^X denotes the power set of X, that is the collection of all subsets of X. A metric for a set X is a function $\mathbf{d} : X \times X \to \mathbb{R}_0^+$ satisfying for any $x_1, x_2, x_3 \in X$ the following properties: $\mathbf{d}(x_1, x_2) = 0$ if and only if $x_1 = x_2$; $\mathbf{d}(x_1, x_2) = \mathbf{d}(x_2, x_1)$; $\mathbf{d}(x_1, x_2) \leq \mathbf{d}(x_1, x_3) + \mathbf{d}(x_3, x_2)$. Given two sets X and Y, $X \setminus Y = \{x \in X \mid x \notin Y\}$. A set X equipped with a binary relation $\preceq \subseteq X \times X$ is a partially ordered set (poset) if for all $x_1, x_2, x_3 \in X$: (reflexivity) $x_1 \preceq x_1$, (antisymmetry) if $x_1 \preceq x_2$ and $x_2 \preceq x_1$ then $x_1 = x_2$, and (transitivity) if $x_1 \preceq x_2$ and $x_2 \preceq x_3$ then $x_1 \preceq x_3$. A maximal element $x_M \in X$ is such that there is no $x \in X$, $x \neq x_M$, such that $x_M \preceq x$. A minimal element $x_m \in X$ is such that there is no $x \in X$, $x \neq x_m$, such that $x \preceq x_m$. Maximum $x_{\max} \in X$ and minimum $x_{\min} \in X$, when they exist, are unique and such that $x_{\min} \preceq x \preceq x_{\max}$, for all $x \in X$. Maximum and minimum are a maximal element and a minimal element of X, respectively, while the converse is not true, in general. In the paper, when clear from the context, we slightly abuse notation by writing X instead of the pair $(X, \preceq)$.

A directed graph G is a pair (V, E) where V is the set of vertices and E is the set of edges. A sink of G is a vertex v for which there is no vertex v' such that $(v, v') \in E$. A root of G is a vertex v for which there is no vertex v' such that $(v', v) \in E$. A cycle is a sequence $v_1 v_2, \ldots, v_{l-1}, v_l$, with $v_i \in V$, $i \in [1; l]$, $(v_i, v_{i+1}) \in E$, $i \in [1; l-1]$ and $v_1 = v_l$. A directed acyclic graph is a directed graph with no cycles.

We recall from e.g. Cassandras and Lafortune (1999) some notions on formal language theory. Let Y be a finite set representing the alphabet. A word over Y of length l is a finite sequence

$$y_1 y_2 \ldots y_l \tag{1}$$

of symbols in Y. The empty word is denoted by ε. The concatenation of two words $y_1 y_2 \ldots y_{l'}$ and $y_{l'+1} y_{l'+2} \ldots y_{l''}$ is the word $y_1 y_2 \ldots y_{l'} y_{l'+1} y_{l'+2} \ldots y_{l''}$. The symbol Y^* denotes the Kleene closure of Y, that is the collection of all words over Y including ε. A language L over Y is a subset of Y^*. The concatenation of two languages L_1 and L_2 is the language $L_1 L_2$ composed of words obtained by concatenating words of L_1 with words of L_2. The Kleene closure of a language L is defined as $L^* = \bigcup_{i \in \mathbb{N}} L^i$, where we set $L^0 = \{\varepsilon\}$.

We now recall the notion of transition system:

Definition 1. A transition system is a tuple

$$T = (X, X_0, U, \longrightarrow, X_m, Y, H), \quad (2)$$

consisting of a set of states X, a set of initial states $X_0 \subseteq X$, a set of inputs U, a transition relation $\longrightarrow \subseteq X \times U \times X$, a set of marked states $X_m \subseteq X$, a set of outputs Y and an output function $H : X \to Y$.

The definition above slightly extends the one of Tabuada (2009) to transition systems with marked states. A transition $(x, u, x') \in \longrightarrow$ of T is denoted by $x \xrightarrow{u} x'$. The evolution of transition systems is captured by the notions of state, input and output runs. Given a sequence

$$x(0) \xrightarrow{u(0)} x(1) \xrightarrow{u(1)} \ldots x(l-1) \xrightarrow{u(l-1)} x(l) \quad (3)$$

of transitions of T with $x(0) \in X_0$, the sequences

$$r_X : x(0)x(1) \ldots x(l),$$
$$r_U : u(0)u(1) \ldots u(l-1), \quad (4)$$
$$r_Y : H(x(0))H(x(1)) \ldots H(x(l)), \quad (5)$$

are called a *state run*, an *input run* and an *output run* of T, respectively. Transition system T in (2) is said to be *finite* if X and U are finite sets, *metric* if Y is equipped with a metric, *deterministic* if for any $x \in X$ and $u \in U$ there exists at most one transition $x \xrightarrow{u} x^+$ and *nondeterministic*, otherwise.

The set of successors of state $x \in X$ with input $u \in U$ is $\mathrm{Post}_u(x) = \{x' \in X \mid x \xrightarrow{u} x'\}$; analogously, the set of predecessors of x with input u is $\mathrm{Post}_u^{-1}(x) = \{x' \in X \mid x' \xrightarrow{u} x\}$. Given $X' \subseteq X$, $\mathrm{Post}_u(X') = \bigcup_{x \in X'} \mathrm{Post}_u(x)$ and $\mathrm{Post}_u^{-1}(X') = \bigcup_{x \in X'} \mathrm{Post}_u^{-1}(x)$.

The *marked input language* (resp. *marked output language*) of T, denoted as $\mathcal{L}_m^u(T)$ (resp. $\mathcal{L}_m^y(T)$), is the collection of all input runs r_U in (4) (resp. output runs r_Y in (5)) such that the corresponding transitions sequence in (3) is with ending state $x_l \in X_m$. In this paper we say that T marks L if $\mathcal{L}_m^y(T) = L$. By following the notions given in Cassandras and Lafortune (1999), a language L over a finite set U is said *regular* if there exists a finite transition system T with input set U such that $L = \mathcal{L}_m^u(T)$. Without loss of generality such transition system T can be assumed as deterministic and with set of initial states as singleton. Moreover,

Proposition 1. *If L_1 and L_2 are regular languages, then $L_1 \cup L_2$, $L_1 \cap L_2$, $L_1 L_2$ and L_1^* are regular languages.*

We can now recall the following

Definition 2. Consider two transition systems

$$T_i = (X_i, X_{0,i}, U_i, \underset{i}{\longrightarrow}, X_{m,i}, Y_i, H_i), \quad i = 1, 2, \quad (6)$$

T_1 is a sub-transition system of T_2, denoted $T_1 \sqsubseteq T_2$, if $X_1 \subseteq X_2$, $X_{0,1} \subseteq X_{0,2}$, $U_1 \subseteq U_2$, $\underset{1}{\longrightarrow} \subseteq \underset{2}{\longrightarrow}$, $X_{m,1} \subseteq X_{m,2}$, $Y_1 \subseteq Y_2$ and $H_1(x) = H_2(x)$ for all $x \in X_1$.

Definition 3. Given T in (2) and $X' \subseteq X$, the sub-transition system of T induced by X' is

$$T|_{X'} = (X', X_0 \cap X', U, \underset{'}{\longrightarrow}, X_m \cap X', Y, H'),$$

where $x \underset{'}{\xrightarrow{u}} x'$ if $x \xrightarrow{u} x'$ and $H'(x) = H(x)$ for all $x \in X'$.

Similarly,

Definition 4. Given T in (2) and $\underset{'}{\longrightarrow} \subseteq \longrightarrow$, the sub-transition system of T induced by $\underset{'}{\longrightarrow} \subseteq \longrightarrow$ and $X'_0 \subseteq X_0$ is

$$T|_{\underset{'}{\longrightarrow}, X'_0} = (X', X'_0, U, \underset{'}{\longrightarrow}, X_m \cap X', Y, H'),$$

where X' is the collection of states in X that are reached by a finite sequence of transitions in $\underset{'}{\longrightarrow}$ starting from an initial state in X'_0.

Consider transition system T in (2).

The accessible part of T, denoted $\mathrm{Ac}(T)$, is the unique maximal[1] sub-transition system T' of T such that for any state x' of T' there exists a state run of T' ending in x'. When $T = \mathrm{Ac}(T)$, it is said accessible. By definition, if T is nonempty, $\mathrm{Ac}(T)$ is accessible.

Define the following recursive equations:

$$X(0) := X_m;$$
$$X(k+1) := X(k) \bigcup_{u \in U} \left\{ \begin{array}{l} x \in \mathrm{Post}_u^{-1}(X(k)) \mid \\ \mathrm{Post}_u(x) \subseteq X(k) \end{array} \right\}.$$

It is readily seen that there exists $k' \in \mathbb{N}$ such that $X(k') = X(k'+1)$ or equivalently, that sequence above admits a fixed point. The co-accessible part of T is defined as $\mathrm{Coac}(T) = T|_{X(k')}$. When $T = \mathrm{Coac}(T)$, it is said co-accessible.

As pointed out in Masciulli et al. (2021) (see Example 2) operators Ac and Coac do not commute. The trim of T, denoted $\mathrm{Trim}(T)$, is defined as $\mathrm{Trim}(T) = \mathrm{Ac}(\mathrm{Coac}(T))$. By definition, $\mathrm{Trim}(T)$, if not empty, is accessible and co-accessible. T is said trim if $T = \mathrm{Trim}(T)$.

For deterministic transition systems, unary operators Ac, Coac and Trim boil down to those in e.g. Cassandras and Lafortune (1999).

We now introduce the following operator on transition systems.

Definition 5. Given a transition system T, $\mathrm{Split}(T)$ is the collection of all sub-transition systems of T that are trim.

$\mathrm{Split}(T)$ can be computed as follows: first compute all sub-transition systems of T, then apply Trim operator to each of them.

We now recall the standard notion of product composition between transition systems.

Definition 6. The product composition between transition systems T_1 and T_2 as in (6), is the transition system T as in (2) where $X = X_1 \times X_2$, $X_0 = X_{0,1} \times X_{0,2}$, $U = U_1 \times U_2$, $\longrightarrow = \underset{1}{\longrightarrow} \times \underset{2}{\longrightarrow}$, $X_m = X_{0,m} \times X_{0,m}$, $Y = Y_1 \times Y_2$ and $H((x_1, x_2)) = (H_1(x_1), H_2(x_2))$, $\forall (x_1, x_2) \in X$.

Product composition is associative, i.e. $(T_1 \times T_2) \times T_3 = T_1 \times (T_2 \times T_3)$.

We conclude this section with the following definition where, for $A \subseteq A_1 \times A_2$, $\mathrm{Proj}'(A, A_i) = \{a_i \in A_i \mid \exists a_{3-i} \in A_{3-i} \text{ s.t. } (a_1, a_2) \in A\}$.

Definition 7. Consider two transition systems T_1 and T_2 as in (6) and a transition system T as in (2), where $T \sqsubseteq T_1 \times T_2$. The projection of T onto T_i is the transition system

$$\mathrm{Proj}(T, T_i) = (X', X'_0, U', \underset{'}{\longrightarrow}, X'_m, Y', H'),$$

where $X' = \mathrm{Proj}'(X, X_i)$, $X'_0 = \mathrm{Proj}'(X, X_{0,i})$, $U' = \mathrm{Proj}'(U, U_i)$, $\underset{'}{\longrightarrow} = \mathrm{Proj}'(\longrightarrow, \underset{i}{\longrightarrow})$, $X'_m = \mathrm{Proj}'(X_m, X_{i,m})$, $Y' = \mathrm{Proj}'(Y, Y_i)$, and $H'(x'_i) = H_i(x'_i)$, for all $x'_i \in X'$.

Given $T \sqsubseteq T_1 \times T_2 \times \cdots \times T_m$, for some transition systems T and T_i, $i \in [1; m]$, the projection $\mathrm{Proj}(T, T_i)$ of T onto T_i, $i \in [1; m]$, can be defined by following the same reasoning as in Definition 7.

[1] Here, maximality is with respect to the pre-order naturally induced by the binary operator $\sqsubseteq$.

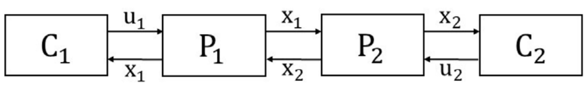

Fig. 1. Decentralized control architecture.

3. Interconnected finite state systems and control problem formulation

The control scheme we consider in this paper is decentralized and illustrated in Fig. 1. We consider a pair of interconnected plants P_1 and P_2 and a pair of controllers C_1 for plant P_1 and C_2 for plant P_2, each controller sharing information only with the corresponding plant.

Each plant P_i, $i = 1, 2$, is modeled as a finite state system and described by the difference inclusion:

$$P_i : \begin{cases} x_i(t+1) \in F_i(x_i(t), x_{3-i}(t), u_i(t)), \\ x_i(0) \in X_{i,0}, x_{3-i}(0) \in X_{3-i,0}, \\ x_i(t) \in X_i, x_{3-i}(t) \in X_{3-i}, u_i(t) \in U_i, t \in \mathbb{N}, \end{cases} \quad (7)$$

where $x_i(t)$ is the state, and $x_{3-i}(t)$ and $u_i(t)$ are the inputs, at time $t \in \mathbb{N}$; $X_i, X_{i,0} \subseteq X_i$ and U_i are the finite sets of states, initial states and inputs, respectively. Input $x_{3-i}(t)$ of P_i is the state of P_{3-i}. Set X_i is equipped with metric $\mathbf{d}_i : X_i \times X_i \to \mathbb{R}_0^+$. Map $F_i : X_i \times X_{3-i} \times U_i \to 2^{X_i}$ is the state transition (possibly partial) map. A state trajectory of P_i of length $l + 1$ is a finite sequence $x_i(0)\, x_i(1) \ldots x_i(l)$ satisfying (7) for some finite sequence of inputs $x_{3-i}(0)\, x_{3-i}(1) \ldots x_{3-i}(l-1)$ and $u_i(0)\, u_i(1) \ldots u_i(l-1)$. Since F_i may be partial, trajectories may be extended to trajectories with arbitrarily large length, or not. Systems P_i are nondeterministic, in general. They become deterministic when for any $(x_i, x_{3-i}, u_i) \in X_i \times X_{3-i} \times U_i$, set $F_i(x_i, x_{3-i}, u_i)$ is either a singleton or empty, and in this last case, the system deadlocks.

Each controller C_i, $i = 1, 2$, is modeled as a finite state system and described by the difference inclusions:

$$C_i : \begin{cases} z_i(t+1) \in G_i(z_i(t), x_i(t)), \\ u_i(t) \in h_i(z_i(t), x_i(t)), \\ z_i(0) \in Z_{i,0}, \\ z_i(t) \in Z_i, u_i(t) \in U_i, t \in \mathbb{N}, \end{cases} \quad (8)$$

where $z_i(t)$, $x_i(t)$ and $u_i(t)$ are the state, the input and the output, at step $t \in \mathbb{N}$, respectively; Z_i, $Z_{i,0} \subseteq Z_i$ and U_i are the finite sets of states, initial states and outputs, respectively. Map $G_i : Z_i \times X_i \to 2^{Z_i}$ is the state transition (possibly partial) map and $h_i : Z_i \times X_i \to 2^{U_i}$ is the output (possibly partial) map. Controller C_i is nondeterministic, in general. It becomes deterministic when for any $(z_i, x_i) \in Z_i \times X_i$, set $G_i(z_i, x_i)$ is either a singleton or empty, and in this last case, C_i deadlocks. When Z_i is a singleton, controller C_i is static and can be simply represented by $C_i : X_i \to 2^{U_i}$. Control action on plant P_i is formalized through the notion of a strategy, as follows.

Definition 8. A (control) strategy S_i for plant P_i is given by a pair (C_i, R_i), where C_i is a controller of the form (8) and $R_i \subseteq X_{i,0} \times Z_{i,0}$ is a relation of initial states between P_i and C_i. The collection of strategies for P_i obtained by considering all possible controllers and relations is denoted by $\mathcal{S}_i$.

Plant P_i under action of strategy $S_i = (C_i, R_i) \in \mathcal{S}_i$ is described by the following difference inclusion:

$$P_i^{S_i} : \begin{cases} (x_i(t+1), z_i(t+1)) \in \phi_i(x_i(t), z_i(t), x_{3-i}(t)), \\ (x_i(0), z_i(0)) \in R_i, \\ (x_i(t), z_i(t)) \in X_i \times Z_i, \\ x_{3-i}(0) \in X_{3-i,0}, \\ x_{3-i}(t) \in X_{3-i}, t \in \mathbb{N}, \end{cases} \quad (9)$$

where state transition map $\phi_i : X_i \times Z_i \times X_{3-i} \to 2^{X_i \times Z_i}$ is defined by:

$$\phi_i(x_i, z_i, x_{3-i}) = \bigcup_{u_i \in h_i(z_i, x_i)} F_i(x_i, x_{3-i}, u_i) \times G_i(z_i, x_i),$$

for all $(x_i, z_i, x_{3-i}) \in X_i \times Z_i \times X_{3-i}$. A state trajectory of $P_i^{S_i}$ is a finite sequence $(x_i(0), z_i(0))\,(x_i(1), z_i(1)) \ldots$ satisfying (9) for some finite sequence of inputs $x_{3-i}(0)\, x_{3-i}(1) \ldots$. Note that the action of strategies S_i on plants P_i may cause a blocking behavior. This happens for example when C_i causes emptiness of $\phi_i(x_i, z_i, x_{3-i})$ for some state $(x_i, z_i, x_{3-i}) \in X_i \times Z_i \times X_{3-i}$, reachable from some initial state.

The interconnection among systems P_1, P_2, C_1 and C_2 is denoted by Σ, depicted in Fig. 1, and is obtained by coupling (7) and (8) for $i = 1, 2$, thus obtaining

$$\Sigma : \begin{cases} (x_1(t+1), z_1(t+1)) \in \phi_1(x_1(t), z_1(t), x_2(t)), \\ (x_2(t+1), z_2(t+1)) \in \phi_2(x_2(t), z_2(t), x_1(t)), \\ (x_1(t), x_2(t), z_1(t), z_2(t)) \in \\ \quad X_1 \times X_2 \times Z_1 \times Z_2, t \in \mathbb{N}. \end{cases} \quad (10)$$

A state trajectory of Σ of length $l + 1$ is a finite sequence $(x_1(0), x_2(0), z_1(0), z_2(0))\,(x_1(1), x_2(1), z_1(1), z_2(1)) \ldots (x_1(l), x_2(l), z_1(l), z_2(l))$ satisfying (10). For later purposes, given a pair of strategies $S_i = (C_i, R_i) \in \mathcal{S}_i$, $i = 1, 2$, we also denote by $\Sigma^{(S_1,S_2)}$ the finite state system obtained by requiring initial states $(x_1(0), x_2(0), z_1(0), z_2(0))$ of Σ to satisfy $(x_1(0), z_1(0)) \in R_1$ and $(x_2(0), z_2(0)) \in R_2$, or equivalently, obtained by coupling (9) for $i = 1$ and $i = 2$.

The control problem we address in this paper is formalized as follows:

Problem 1. Consider plants P_i, specifications Q_i described by regular languages over alphabet X_i, i.e. $Q_i \subseteq (X_i)^*$, and desired accuracies $\theta_i \in \mathbb{R}_0^+$, $i = 1, 2$. Design strategies $S_i \in \mathcal{S}_i$ such that $P_i^{S_i}$ satisfy Q_i within a desired accuracy θ_i, $i = 1, 2$, i.e. for any state trajectory $(x_1(.), x_2(.), z_1(.), z_2(.))$ of $\Sigma^{(S_1,S_2)}$, with length $l+1$, there exist two words $q_0^i\, q_1^i \ldots q_l^i \in Q_i$, $i = 1, 2$, such that the following inequalities hold:

$$\mathbf{d}_i(x_i(t), q_t^i) \leq \theta_i, \forall t \in [0; l], i = 1, 2. \quad (11)$$

In this paper, we provide a characterization of all solutions to the problem above. Although in this paper we are considering dynamic controllers, there are cases where Problem 1 admits solution by using static controllers, i.e. controllers of the form $C_i : X_i \to 2^{U_i}$, which are easier to be designed and also implemented. In the examples presented in this paper we will see both cases of dynamic and static controllers. Problem 1 is a decentralized control problem with local specifications. When designing $S_i \in \mathcal{S}_i$ to enforce Q_i on P_i, it may be restrictive to treat the external input x_{3-i} as an unknown disturbance, as illustrated in the following example.

Example 1. Consider P_1 with $X_1 = \{0, 1, 2, 3, 4, 5\}$, $X_{1,0} = \{0, 2\}$, $U_1 = \{a, b, c\}$, P_2 with $X_2 = \{0', 1', 2', 3', 4', 5'\}$, $X_{2,0} = \{0', 1'\}$, $U_2 = \{a', b'\}$, and transition maps F_1 and F_2 depicted in Fig. 2. Specifications are given by languages

$$\begin{aligned} Q_1 &= \{0\,1\,0\}\{1\,0\}^* \cup \{0\,1\,5,\ 0\,1\,4,\ 0\,2\,4, 0\,2\,3, 2\,4\}, \\ Q_2 &= \{0'\}\{1'\}^* \cup \{0'\,1'\,5',\ 0'\,1'\,4',\ 0'\,2'\,4',\ 0'\,2'\,3',\ 1'5'\}, \end{aligned} \quad (12)$$

over alphabets X_1 and X_2, respectively. Since $\{0\,1\,0\}$, $\{1\,0\}$, $\{0\,1\,5,\ 0\,1\,4,\ 0\,2\,4, 0\,2\,3, 2\,4\}$, $\{0'\}$, $\{1'\}$ and $\{0'\,1'\,5',\ 0'\,1'\,4',\ 0'\,2'\,4',\ 0'\,2'\,3',\ 1'5'\}$ are regular languages, by Proposition 1, languages Q_1 and Q_2 in (12) are regular. If P_i treats x_{3-i} as an unknown disturbance, there are no control strategies enforcing Q_1 and Q_2. In fact, if P_1 starts from 2 and P_2 starts from $0'$, P_1

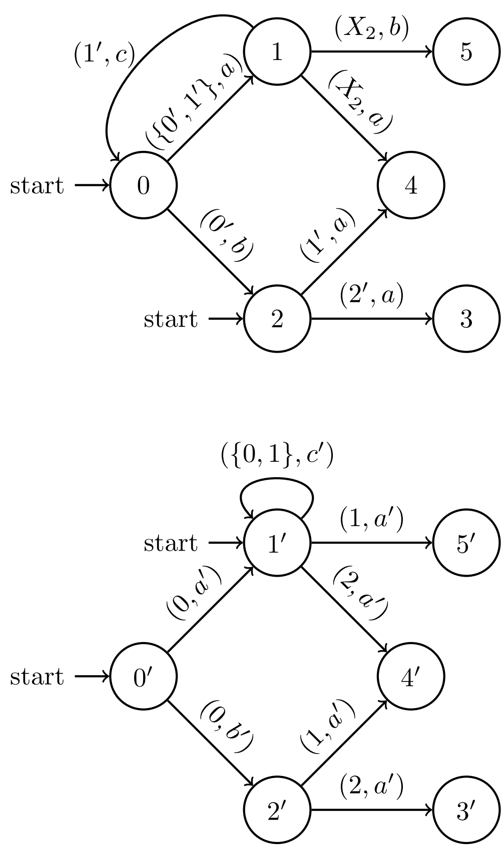

Fig. 2. Plant P_1 (upper panel) and plant P_2 (lower panel) of Example 1.

blocks and word $2 \notin Q_1$. On the other hand, it is readily seen that for example, strategies $S_i = (C_i, R_i)$, $i = 1, 2$, with $R_1 = \{0\}$, $R_2 = \{0'\}$ and static controllers C_1 and C_2 specified by $C_1(0) = \{b\}$, $C_1(2) = \{a\}$, $C_2(0') = \{b'\}$, $C_2(2') = \{a'\}$, enforce words $0\,2\,3 \in Q_1$ and $0'\,2'\,3' \in Q_2$ and hence, solve Problem 1 for any metric chosen and with accuracies $\theta_1 = \theta_2 = 0$.

This observation prompted us to explore alternative approaches to derive solutions to Problem 1 that are inspired by game theory.

4. Games and equilibria

We consider a game with two players P_i, $i = 1, 2$. Actions of P_i are formalized by strategies $S_i \in \mathcal{S}_i$. Reward functions of P_i are formalized by the notion of parts of specification enforced, as follows:

Definition 9. For $i = 1, 2$, the part of the specification Q_i enforced on P_i by strategy $S_i = (C_i, R_i) \in \mathcal{S}_i$, given strategy $S_{3-i} = (C_{3-i}, R_{3-i}) \in \mathcal{S}_{3-i}$, denoted by $\mathcal{Q}_i(S_i, S_{3-i})$, is the collection of words $q_0\, q_1 \, \ldots \, q_l \in Q_i$ for which there exists a state trajectory $(x_1(.), x_2(.), z_1(.), z_2(.))$ of $\Sigma^{(S_1,S_2)}$ with length $l + 1$, such that the following inequality holds:

$$\mathbf{d}_i(x_i(t), q_t) \leq \theta_i, \forall t \in [0; l]. \tag{13}$$

We can now introduce the following definition.

Definition 10. The pair $(S_1, S_2) \in \mathcal{S}_1 \times \mathcal{S}_2$ is an equilibrium if $\mathcal{Q}_1(S_1, S_2) \neq \varnothing$ and $\mathcal{Q}_2(S_2, S_1) \neq \varnothing$. We denote by $\mathcal{E}$ the collection of equilibria in Σ.

By Definition 10, $(S_1, S_2) \in \mathcal{E}$ if and only if it solves Problem 1. In the definition above, each plant P_i chooses a strategy S_i, without knowing the state of the other plant P_{3-i}. Even with this partial information setting, pair of strategies (S_1, S_2) provide a solution to the problem at hand. Hence, strategies S_1 and S_2 can be thought of as being in balance, hence the term “equilibrium”. In the definition above we skip dependency of $\mathcal{E}$ on θ_1 and θ_2 for notational simplicity. It is easy to see that if $(S_1, S_2) \in \mathcal{E}$ then (S_1, S_2) is also an equilibrium with respect to any accuracies $\theta'_1 \geq \theta_1$ and $\theta'_2 \geq \theta_2$.

In the sequel we focus on characterizing and computing equilibria. We start by illustrating some properties of $\mathcal{E}$.

The set $\mathcal{E}$ may be empty, in general. If $\mathcal{E} \neq \varnothing$, it has infinite cardinality. Indeed, let $(S_1, S_2) \in \mathcal{E}$ with $S_i = (C_i, R_i)$. For $i = 1, 2$, consider strategies $S'_i = (C'_i, R'_i)$, where C'_i is obtained by coupling equations in (8) with “spurious” dynamics not influencing $z_i(t)$, e.g. $z'_i(t+1) = z'_i(t), z'_i(t) \in Z'_i$, $t \in \mathbb{N}$, and $R'_i = R_i \times Z'_i$. Then, $S'_i \neq S_i$ and it is readily seen that $(S'_1, S'_2) \in \mathcal{E}$. By coupling equations in (8) with an arbitrarily large number of “spurious” dynamics, we obtain an arbitrarily large number of “artificious” new equilibria. More in general, starting from one equilibrium (S_1, S_2), one can appropriately construct an arbitrarily large number of pairs of strategies $(S'_1, S'_2) \in \mathcal{S}_1 \times \mathcal{S}_2$ that are equivalent via alternating bisimulation (see e.g. Alur et al. (1998), Pola and Tabuada (2009) and Tabuada (2009)) to (S_1, S_2). Then, by invoking the so-called congruence property, see e.g. Clarke et al. (1999), for $i = 1, 2$, $P_i^{S_i}$ and $P_i^{S'_i}$ are equivalent via alternating bisimulation from which, $\mathcal{Q}_i(S_i, S_{3-i}) = \mathcal{Q}_i(S'_i, S'_{3-i})$ and hence, $(S'_1, S'_2) \in \mathcal{E}$.

Moreover, it is readily seen that $\mathcal{E}$ is a partially ordered set (poset) when equipped with binary relation $\preceq$ defined by

$$(S_1, S_2) \preceq (S'_1, S'_2),$$

if $\mathcal{Q}_1(S_1, S_2) \subseteq \mathcal{Q}_1(S'_1, S'_2)$ and $\mathcal{Q}_2(S_2, S_1) \subseteq \mathcal{Q}_2(S'_2, S'_1)$. When $(S_1, S_2) \preceq (S'_1, S'_2)$, strategy (S_1, S_2) is said to be dominated by (S'_1, S'_2). The following example shows that in general, the poset $\mathcal{E}$ does not have a maximum.

Example 2. Consider P_1, P_2, Q_1 and Q_2 of Example 1 and suppose $\theta_1 = \theta_2 = 0$. Consider static controller C_1 specified by $C_1(0) = \{a\}$, $C_1(1) = \{c\}$, and set of initial states $R_1 = \{0\}$. Consider also static controller C_2 specified by $C_2(0') = \{a'\}$, $C_2(1') = \{c'\}$, and set of initial states $R_2 = \{0'\}$. By setting $S_1 = (C_1, R_1)$ and $S_2 = (C_2, R_2)$, it is easy to see that

$$\begin{aligned} &\mathcal{Q}_1(S_1, S_2) = \{0\,1\,0\}\{1\,0\}^*, \\ &\mathcal{Q}_2(S_2, S_1) = \{0'\}\{1'\}^*. \end{aligned} \tag{14}$$

Hence, $(S_1, S_2) \in \mathcal{E}$. Consider now static controller C'_1 specified by $C'_1(0) = \{a, b\}$, $C'_1(1) = \{a, b\}$, $C'_1(2) = \{a\}$ and set of initial states $R'_1 = \{0\}$. Consider also static controller C'_2 specified by $C'_2(0') = \{a', b'\}$, $C'_2(1') = \{a'\}$, $C'_2(2') = \{a'\}$, and set of initial states $R'_2 = \{0'\}$. By setting $S'_1 = (C'_1, R'_1)$ and $S'_2 = (C'_2, R'_2)$, it is easy to see that

$$\begin{aligned} &\mathcal{Q}_1(S'_1, S'_2) = \{0\,1\,5,\ 0\,1\,4,\ 0\,2\,4, 0\,2\,3\}, \\ &\mathcal{Q}_2(S'_2, S'_1) = \{0'\,1'\,5',\ 0'\,1'\,4',\ 0'\,2'\,4',\ 0'\,2'\,3'\}. \end{aligned} \tag{15}$$

Hence, $(S'_1, S'_2) \in \mathcal{E}$. From (14) and (15), neither $(S_1, S_2) \preceq (S'_1, S'_2)$ nor $(S'_1, S'_2) \preceq (S_1, S_2)$. We now claim that there is no equilibrium, say (S''_1, S''_2) with $S''_i = (C''_i, R''_i)$, such that $(S_1, S_2) \preceq (S''_1, S''_2)$ and $(S'_1, S'_2) \preceq (S''_1, S''_2)$, a necessary condition for the existence of the maximum of $\mathcal{E}$. For

$$\mathcal{Q}_1(S_1, S_2) \cup \mathcal{Q}_1(S'_1, S'_2) \subseteq \mathcal{Q}_1(S''_1, S''_2)$$

to hold, we need $C''_1(0) = \{a, b\}$ and $C''_1(1) = \{a, b, c\}$; analogously, for

$$\mathcal{Q}_2(S_2, S_1) \cup \mathcal{Q}_2(S'_2, S'_1) \subseteq \mathcal{Q}_2(S''_2, S''_1)$$

to hold, we need $C_2''(0') = \{a', b'\}$ and $C_2''(1') = \{a', c'\}$. Sets of initial states in S_1'' and S_2'' are $R_1'' = \{0\}$ and $R_2'' = \{0'\}$. Suppose: P_1 picks $a \in C_1''(x_1(0))$ at step $t = 0$ and picks $c \in C_1''(x_1(1))$ at step $t = 1$; P_2 picks $b' \in C_2''(x_2(0))$ at step $t = 0$ and picks $a' \in C_2''(x_2(1))$ at step $t = 1$. At step $t = 1$, P_1 needs state $x_2(1) = 1'$ of P_2 to evolve. Since $x_2(1) = 2'$, P_1 blocks and trajectory 0 1 obtained is not in Q_1. Hence, strategy (S_1'', S_2'') does not exist and the claim is proven.

Since poset $\mathcal{E}$ has no maximum in general, in the sequel we study and characterize all equilibria.

We start by introducing a special class of equilibria:

Definition 11. An equilibrium $(S_1, S_2) \in \mathcal{E}$ is Nash if
- $\mathcal{Q}_1(S_1', S_2) \subseteq \mathcal{Q}_1(S_1, S_2)$, $\forall S_1' \in \mathcal{S}_1$;
- $\mathcal{Q}_2(S_2', S_1) \subseteq \mathcal{Q}_2(S_2, S_1)$, $\forall S_2' \in \mathcal{S}_2$.

We denote by $\mathcal{N}$ the collection of Nash equilibria in Σ.

By the definition above, whenever only player P_1 deviates from a Nash equilibrium (S_1, S_2) by selecting a different strategy $S_1' \neq S_1$, while P_2 maintains strategy S_2, the corresponding part of the specification Q_1 enforced may reduce, and vice versa. This corresponds *mutatis mutandis* to the classical notion of Nash equilibrium, as pointed out in the following:

Remark 1. In static two–players games, a collection of strategies in some sets V_1 and V_2 and a pair of reward functions $J_1 : V_1 \times V_2 \to \mathbb{R}$ and $J_2 : V_2 \times V_1 \to \mathbb{R}$ are given. A Nash equilibrium for this game is given by a pair $(v_1, v_2) \in V_1 \times V_2$ for which $J_1(v_1', v_2) \leq J_1(v_1, v_2)$ for all $v_1' \in V_1$ and $J_2(v_2', v_1) \leq J_2(v_2, v_1)$ for all $v_2' \in V_2$. In our framework, reward functions J_i correspond to $\mathcal{Q}_i$ ranging in the poset 2^{Q_i} (with partial order $\subseteq$) rather than in the totally ordered set $\mathbb{R}$ as in the case of J_i.

As for $\mathcal{E}$, it is easy to see that $\mathcal{N}$, when non-empty, has infinite cardinality. We also introduce the notion of dominant equilibrium.

Definition 12. An equilibrium $(S_1, S_2) \in \mathcal{E}$ is dominant if
- $\mathcal{Q}_1(S_1', S_2') \subseteq \mathcal{Q}_1(S_1, S_2)$, $\forall S_1' \in \mathcal{S}_1$ and $\forall S_2' \in \mathcal{S}_2$;
- $\mathcal{Q}_2(S_2', S_1') \subseteq \mathcal{Q}_2(S_2, S_1)$, $\forall S_2' \in \mathcal{S}_2$ and $\forall S_1' \in \mathcal{S}_1$.

We denote by $\mathcal{D}$ the collection of dominant equilibria in Σ.

As for $\mathcal{E}$ and $\mathcal{N}$, it is easy to see that $\mathcal{D}$, when non-empty, has infinite cardinality. A dominant equilibrium is also Nash while the converse is not true, in general. In fact, it is readily seen that equilibria (S_1, S_2) and (S_1', S_2') in Example 2 are Nash but not dominant. The following example clarifies further connections between Nash and dominant equilibria.

Example 3. Consider P_1 with $X_1 = \{0, 1, 2, 3, 4\}$, $X_{1,0} = \{0\}$, $U_1 = \{a, b\}$, P_2 with $X_2 = \{0', 1', 2', 3', 4'\}$, $X_{2,0} = \{0'\}$, $U_2 = \{a', b', c'\}$, and transition maps F_1 and F_2 depicted in Fig. 3. Suppose $\theta_1 = \theta_2 = 0$. Consider a pair of regular language specifications $Q_1 = \{0\,1\,1,\ 0\,2\,3\,4,\ 0\,2\,3\,4\,5\}$ and $Q_2 = \{0'\,0'\,1',\ 0'\,0'\,2'\,3',\ 0'\,0'\,2'\,3'\,4'\}$. Consider strategies $S_1 = (C_1, R_1)$ and $S_2 = (C_2, R_2)$ defined by $R_1 = \{0\}$, $R_2 = \{0'\}$, $C_1(0) = \{a, b\}$, $C_1(1) = \{a\}$, $C_1(2) = C_1(3) = \{b\}$, $C_2(0') = \{a'\}$ and $C_2(2') = \{b'\}$. It is easy to see that $\mathcal{Q}_1(S_1, S_2) = \{0\,1\,1,\ 0\,2\,3\,4\}$, $\mathcal{Q}_2(S_2, S_1) = \{0'\,0'\,1',\ 0'\,0'\,2'\,3'\}$, and hence, (S_1, S_2) is an equilibrium and it is dominant. Consider now strategies $S_1' = (C_1', R_1')$ and $S_2' = (C_2', R_2')$ defined by $R_1' = \{0\}$, $R_2' = \{0'\}$, $C_1(0) = C_1(1) = \{a\}$ and $C_2(0') = C_2(1') = \{a'\}$. We get $\mathcal{Q}_1(S_1', S_2') = \{0\,1\,1\}$ and $\mathcal{Q}_2(S_2', S_1') = \{0'\,0'\,1'\}$ from which $(S_1', S_2') \in \mathcal{E}$. Suppose P_1 deviates from strategy S_1 by applying input b when in state 0 and that P_2 applies S_2. P_1 and P_2 then reach states 2 and $2'$, respectively. When in $2'$, plant P_2 blocks because controller C_2' is not defined in state $2'$. When in 2, plant P_1 reaches state 3 and then blocks. By deviation from strategy S_1, both P_1 and

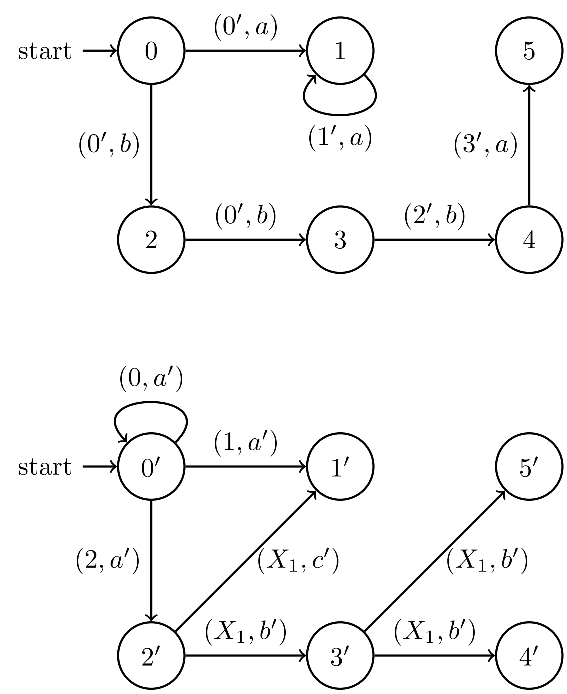

Fig. 3. Plant P_1 (upper panel) and plant P_2 (lower panel) of Example 3.

P_2 do not enforce any word in Q_1 and Q_2, respectively. Conversely, any strategy for P_2 other than S_2' is such that it enforces either no words or the same part of specification as S_2'. As a consequence, (S_1', S_2') is Nash. Moreover, it is dominated by (S_1, S_2), i.e. $(S_1', S_2') \preceq (S_1, S_2)$.

We conclude this section with an example given in the non-deterministic case, for accuracies $\theta_i \geq 0$ and with dynamic controllers.

Example 4. Consider interconnected systems P_1 and P_2 given by $X_1 = \{0, 1, 2, 3, 4, 5\}$, $X_{1,0} = \{0, 3\}$, $U_1 = \{a, b\}$, $X_2 = \{0', 1', 2', 3', 4', 5'\}$, $X_{2,0} = \{0', 4'\}$, $U_2 = \{a', b'\}$ and transition maps F_1 and F_2 depicted in Fig. 4. Metric $\mathbf{d}_1$ is defined by $\mathbf{d}_1(m, n) = |m - n|$, $m, n \in X_1$, and metric $\mathbf{d}_2$ by $\mathbf{d}_2(m', n') = |m' - n'|$, $m', n' \in X_2$. P_1 is nondeterministic while P_2 is deterministic. Consider a pair of specifications $Q_1 = \{(0\,1\,2\,1)^*(0\,1\,2\,1\,0),\ 3\,4,\ 3\,5,\ 3\,5\,4,\ 3\,5\,5\}$ and $Q_2 = \{(0'\,1'\,2'\,3')^*(0'\,1'\,2'\,3'\,0'),\ 4'\,5',\ 4'\,5'\,5',\ 4'\,5'\,5'\,5'\}$. Suppose first $\theta_1 = \theta_2 = 0$. We get:

- Define dynamic controller C_1 as in Fig. 5 (upper label in each circle is the state and lower label is its output) and $R_1 = \{(0, 0'')\}$. Define static controller C_2 by $C_2(0') = C_2(1') = C_2(2') = C_2(3') = \{a'\}$ and $R_2 = \{0'\}$. Define strategies $S_i = (C_i, R_i) \in \mathcal{S}_i$, $i = 1, 2$. Since $\mathcal{Q}_1(S_1) = \{(0\,1\,2\,1)^*(0\,1\,2\,1\,0)\} \neq \varnothing$ and $\mathcal{Q}_2(S_2) = \{(0'\,1'\,2'\,3')^*(0'\,1'\,2'\,3'\,0')\} \neq \varnothing$, then $(S_1, S_2) \in \mathcal{E}$.

- Define static controller C_1' by $C_1'(3) = \{b\}$, $C_1'(5) = \{a\}$ and $R_1' = \{3'\}$. Define static controller C_2' by $C_2'(4') = C_2'(5') = \{a'\}$ and $R_2' = \{4'\}$. Define $S_i' = (C_i', R_i') \in \mathcal{S}_i$, $i = 1, 2$. Since $\mathcal{Q}_1(S_1') = \{3\,4,\ 3\,5,\ 3\,5\,4\} \neq \varnothing$ and $\mathcal{Q}_2(S_2') = \{4'\,5',\ 4'\,5'\,5'\} \neq \varnothing$, then $(S_1', S_2') \in \mathcal{E}$.

- By using arguments similar to those in Example 2, it is easy to see that (S_1, S_2) and (S_1', S_2') are Nash but not dominant.

Suppose now $\theta_1 = \theta_2 = 1$. Equilibria (S_1, S_2) and (S_1', S_2') (computed for $\theta_1 = \theta_2 = 0$) are also equilibria for $\theta_1 = \theta_2 = 1$. In particular, since $|5 - 4| = 1$, we get $\mathcal{Q}_1(S_1') = \{3\,4,\ 3\,5,\ 3\,5\,4,\ 3\,5\,5\}$ that is larger than $\mathcal{Q}_1(S_1')$ in the previous case, while $\mathcal{Q}_2(S_2')$ is the same.

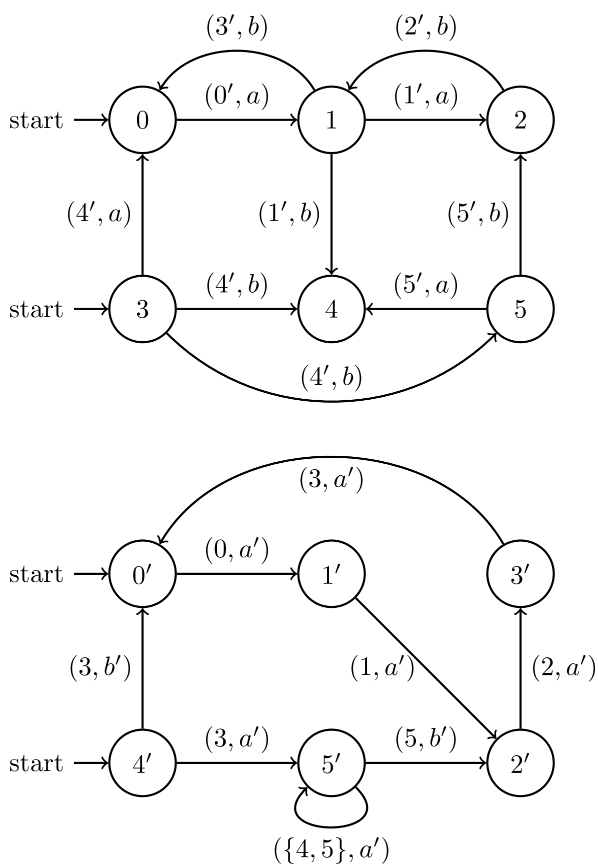

Fig. 4. Plant P_1 (upper panel) and plant P_2 (lower panel) of Example 4.

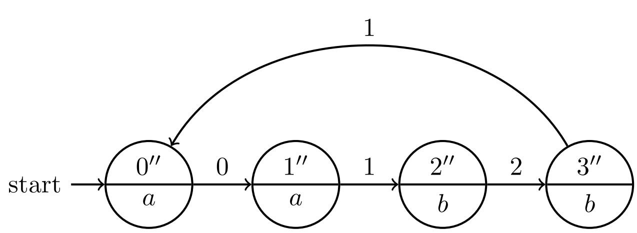

Fig. 5. Controller C_1 of Example 4.

5. Results

This section is organized as follows. In Section 5.1 we provide an algorithm that computes equilibria $(S_1, S_2) \in \mathcal{E}$ and in Section 5.2, a characterization of, and tools for computing, equilibria, Nash and dominant equilibria.

5.1. Computing equilibria

We start by associating a transition system T_{Q_i} to specification Q_i, called dual transition system (see Definition 6 in Pola et al. (2018)), specified by

$$T_{Q_i} = (X_{q,i}, X_{q,0,i}, U_{q,i}, \xrightarrow[q,i]{}, X_{q,m,i}, Y_{q,i}, H_{q,i}), \tag{16}$$

with $Y_{q,i} = X_i$, and satisfying the following properties:
(*i*) T_{Q_i} is finite, trim and metric with metric $\mathbf{d}_i$;
(*ii*) the marked output language $\mathcal{L}^y_m(T_{Q_i})$ of T_{Q_i} coincides with Q_i;
(*iii*) T_{Q_i} is of minimal cardinality in the set of states.

When specialized to Finite State Automata (FSA), transition system T_{Q_i} coincides with the dual FSA (Gol et al., 2014). Construction of T_{Q_i} can be done by resorting to standard algorithms available in the literature, see e.g. Lawson (2004). Automatic tools for constructing it are also well known, see e.g. Caugherty (1990). Algorithms for computing minimal transition systems preserving equivalence of languages are well studied in the literature, see e.g. Cassandras and Lafortune (1999).

Transition system T_{Q_i} satisfies the following property:

Proposition 2. *For any word $w_i \in \mathcal{L}^y_m(T_{Q_i}) = Q_i$ there exists a unique state run of T_{Q_i} with output run w_i.*

Proof. By definition of regular languages, there exists a deterministic finite transition system T_i with set of initial state as singleton, such that $Q_i = \mathcal{L}^u_m(T_i)$. Properties of T_i imply that for any word $w_i \in \mathcal{L}^u_m(T_i)$ there exists a unique state run of T_i with input run w_i. By definition of T_{Q_i}, for any state run r_X of T_i there exists a unique state run r'_X of T_{Q_i} such that input run of r_X coincides with output run of r'_X and, vice versa. Hence, the result follows. □

Consider now $\text{Split}(T_{Q_1})$ and $\text{Split}(T_{Q_2})$, see Definition 5. It is readily seen that

Proposition 3. *For $i = 1, 2$, pair $(\text{Split}(T_{Q_i}), \sqsubseteq)$ is a poset.*

Proposition 4. *Pair $(\text{Split}(T_{Q_1}) \times \text{Split}(T_{Q_2}), \sqsubseteq')$, where $\sqsubseteq'$ is defined by $(T_1, T_2) \sqsubseteq' (T'_1, T'_2)$, if $T_1 \sqsubseteq' T'_1$ and $T_2 \sqsubseteq' T'_2$, is a poset.*

Proofs of the two propositions above are omitted, since the ordering relation $\sqsubseteq$ satisfies reflexivity, anti-symmetry and transitivity properties, as requested for posets.

Computation of equilibria is based on Algorithm 1 that takes as input, plants P_1 and P_2 and transition systems $T'_{Q_1} \in \text{Split}(T_{Q_1})$ and $T'_{Q_2} \in \text{Split}(T_{Q_2})$. Transition systems T'_{Q_1} and T'_{Q_2}, are finite, trim, with output sets X_1 and X_2, respectively, metric with metrics $\mathbf{d}_1$ and $\mathbf{d}_2$, respectively (same metrics as P_1 and P_2), and mark parts of, or, all specification Q_i, i.e. $\mathcal{L}^y_m(T'_{Q_1}) \subseteq Q_1$ and $\mathcal{L}^y_m(T'_{Q_2}) \subseteq Q_2$. Algorithm 1 gives as output a collection of pairs of strategies (S_1, S_2). Transition system T^c in lines 5 and 6 embeds information about plants P_1 and P_2 and transition systems T'_{Q_1} and T'_{Q_2}. Note that only quadruples $(x_1, x_{q,1}, x_2, x_{q,2})$ for which $\mathbf{d}_1(x_1, H'_{q,1}(x_{q,1})) \leq \theta_1$ and $\mathbf{d}_2(x_2, H'_{q,2}(x_{q,2})) \leq \theta_2$ are included in X^c. Line 7 extracts from T^c its accessible and co-accessible part, by using Trim operator. Lines 10–18 extract from a transition system $T^c(j) \in \text{Split}(T^c)$ a transition system for which strategies can be designed to deal with the nondeterminism of the plants. Lines 19–23 extract from $\xrightarrow[c,j]{}$, transitions where input u_i that enforces specification Q_i on P_i, only depends on states x_i and z_i. Transition system $T^c(j)$ in line 23, contains all information from which, strategies can be derived. Lines 24–26 output strategies $S_i = (C_i, R_i)$.

Remark 2. Note that $T^c \sqsubseteq T_{P_1} \times T_{Q_1} \times T_{P_2} \times T_{Q_2}$ and $T^c(j) \sqsubseteq T_{P_1} \times T_{Q_1} \times T_{P_2} \times T_{Q_2}$ where T_{P_i} is the transition system representation of P_i, i.e. $T_{P_i} = (X_i, X_{i,0}, U_i \times X_{3-i}, \xrightarrow[i]{}, X_i, X_i, H_i)$, where $x_i \xrightarrow[i]{u_i, x_{3-i}} x_i^+$, if $x_i^+ \in F_i(x_i, x_{3-i}, u_i)$, and $H_i(x_i) = x_i$, $x_i \in X_i$, $i = 1, 2$.

The following result holds:

Proposition 5. *Algorithm 1 terminates in a finite number of steps.*

Proof. Number of iterations to define $\xrightarrow[c]{}$ in line 5–6 is upper bounded by

$$\text{card}(X_1 \times X'_{q,1,0} \times X_2 \times X'_{q,2,0})^2 \text{card}(U_1)\text{card}(U_2). \tag{17}$$

1 **input:**
2 plants P_i;
3 specification transition systems $T'_{Q_i} = (X'_{q,i}, X'_{q,0,i}, U_{q,i}, \xrightarrow[\prime,q,i]{}, X'_{q,m,i}, X_i, H'_{q,i}) \in \mathrm{Split}(T_{Q_i})$, $i = 1, 2$;
4 **init:**
5 $X^c := \{(x_1, x_{q,1}, x_2, x_{q,2}) \in X_1 \times X'_{q,1} \times X_2 \times X'_{q,2} \mid \mathbf{d}_1(x_1, H'_{q,1}(x_{q,1})) \leq \theta_1 \wedge \ \mathbf{d}_2(x_2, H'_{q,2}(x_{q,2}))) \leq \theta_2\}$;
$X^c_0 := X^c \cap (X_{1,0} \times X'_{q,1,0} \times X_{2,0} \times X'_{q,2,0})$; $U^c := U_1 \times U_2$; $X^c_m := \{(x_1, x_{q,1}, x_2, x_{q,2}) \in X^c \mid x_{q,1} \in X'_{q,m,} \wedge x_{q,2} \in X'_{q,m,2}\}$; $Y^c := X_1 \times X_2$;
$H^c(x_1, x_{q,1}, x_2, x_{q,2}) := H'_{q,1}(x_{q,1}) \times H'_{q,2}(x_{q,2})$, $\forall (x_1, x_{q,1}, x_2, x_{q,2}) \in X^c$;
6 Define $\xrightarrow[c]{}$ by $(x_1, x_{q,1}, x_2, x_{q,2}) \xrightarrow[c]{(u_1,u_2)} (x_1^+, x_{q,1}^+, x_2^+, x_{q,2}^+)$ if

$[x_1^+ \in F_1(x_1, x_2, u_1)] \wedge [x_{q,1} \xrightarrow[\prime,q,1]{} x_{q,1}^+] \wedge [x_2^+ \in F_2(x_2, x_1, u_2)] \wedge [x_{q,2} \xrightarrow[\prime,q,2]{} x_{q,2}^+]$;

7 $T^c := (X^c, X^c_0, U^c, \xrightarrow[c]{}, X^c_m, Y^c, H^c)$; $T^c := \mathrm{Trim}(T^c)$;
8 **main:**
9 $\mathrm{Split}(T^c) = \{T^c(1), T^c(2), \ldots\}$;
10 **foreach** $T^c(j) := (X^c(j), X^c_0(j), U^c(j), \xrightarrow[c,j]{}, X^c_m(j), Y^c(j), H^c(j)) \in \mathrm{Split}(T^c)$ **do**
11 $\xrightarrow[temp]{} := \varnothing$;
12 **foreach** $x = (x_1, x_{q,1}, x_2, x_{q,2}) \xrightarrow[c,j]{(u_1,u_2)} x^+$ **do**
13 **if** $x' = (x'_1, x'_{q,1}, x'_2, x'_{q,2}) \in X^c(j)$ *for all*

$x'_1 \in F_1(x_1, x_2, u_1), \quad x_{q,1} \xrightarrow[c]{} x'_{q,1}, \quad x'_2 \in F_2(x_2, x_1, u_2), \quad x'_{q,2} \xrightarrow[c]{} x'_{q,2}$

then
14 add all $x \xrightarrow[c]{(u_1,u_2)} x'$ to $\xrightarrow[temp]{}$;
15 **end**
16 **end**
17 $X'_0 = (X^c_0(j) \cap X^c_m(j)) \cup \{x \in X^c_0(j) \mid x \xrightarrow[c,j]{u} x'\}$;
18 $T^c(j) := (X^c(j), X^c_0(j), U^c(j), \xrightarrow[c,j]{}, X^c_m(j), Y^c(j), H^c(j)) = \mathrm{Trim}(T^c(j)|_{\xrightarrow[temp]{}, X'_0})$;
19 **foreach** $x_c = (x_1, x_{q,1}, x_2, x_{q,2}) \in X^c(j)$ **do**
20
$U_i(x_c) := \{u_i \in U_i \mid \exists u_{3-i} \in U_{3-i} \text{ s.t. } x_c \xrightarrow[c,j]{(u_1,u_2)} x_c^+\}, \quad i = 1, 2,$
$U'_i(x_c) := \bigcap_{x'_c = \left(x_1, x_{q,1}, x'_2, x'_{q,2}\right) \in X^c(j)} U_i(x'_c), \quad i = 1, 2,$

21 **end**
22 $\xrightarrow[c,j]{} := \left\{x_c \xrightarrow[c,j]{(u_1,u_2)} x_c^+ \mid u_1 \in U'_1(x_c) \wedge u_2 \in U'_2(x_c)\right\}$;
23 $T^c(j) := (X^c(j), X^c_0(j), U^c(j), \xrightarrow[c,j]{}, X^c_m(j), Y^c(j), H^c(j))$; $T^c(j) := \mathrm{Trim}(T^c(j))$;
24 construct strategies $S_i(j) = (C_i^j, R_i^j)$, $i = 1, 2$, where:
25 controllers C_i^j, $i = 1, 2$, are defined by

$Z_i^j := \{(x_i, x_{q,i}) \in X_i \times X'_{q,i} \mid \exists (x_{3-i}, x_{q,3-i}) \in X_{3-i} \times X'_{q,3-i} \text{ s.t. } (x_i, x_{q,i}, x_{3-i}, x_{q,3-i}) \in X^c(j)\}$;
$Z_{i,0}^j := \{(x_i, x_{q,i}) \in X_i \times X'_{q,i} \mid \exists (x_{3-i}, x_{q,3-i}) \in X_{3-i} \times X'_{q,3-i} \text{ s.t. } (x_i, x_{q,i}, x_{3-i}, x_{q,3-i}) \in X^c_0(j)\}$;
$G_i^j((x_i, x_{q,i}), x_i) := \left\{ (x_i^+, x_{q,i}^+) \in Z_i^j \mid \exists\, (x_i, x_{q,i}, x_{3-i}, x_{q,3-i}) \xrightarrow[c,j]{(u_i,u_{3-i})} (x_i^+, x_{q,i}^+, x_{3-i}^+, x_{q,3-i}^+) \right\}$;
$h_i^j((x_i, x_{q,i}), x_i) := \left\{u_i \in U_i \mid \exists\, (x_i, x_{q,i}, x_{3-i}, x_{q,3-i}) \xrightarrow[c,j]{(u_i,u_{3-i})} (x_i^+, x_{q,i}^+, x_{3-i}^+, x_{q,3-i}^+)\right\}$;

26 relations R_i^j, $i = 1, 2$ are defined by $R_i^j := \{(x'_i, (x_i, x_{q,i})) \in X_i \times Z_{i,0}^j \mid x'_i = x_i\}$.
27 **end**

Algorithm 1. Computation of equilibria.

Number of iterations in lines 11–17 and 18–22 is upper bounded by (17), as well. Hence, the result follows. □

Remark 3. Space complexity of Algorithm 1 is upper bounded by (17). On the other hand, computation of X^c and $\xrightarrow[c]{}$ can be done by starting from initial states of T^c and by adding to X^c and $\xrightarrow[c]{}$ only feasible next states and transitions. This approach, based on on-the-fly algorithms, see e.g. Borri et al. (2012), Gerth et al. (1996) and Pola et al. (2012), may lead to computational space savings. Same reasoning applies to computational time.

The following result shows that outputs of Algorithm 1 are either empty or pairs of strategies (S_1, S_2) that are equilibria.

Theorem 1. *Suppose Algorithm 1, with inputs P_i and T'_{Q_i}, $i = 1, 2$, outputs non-empty strategies S_1 and S_2. Then, $(S_1, S_2) \in \mathcal{E}$.*

Proof. Consider any state trajectory $(x_1(.), x_2(.), z_1(.), z_2(.))$ of $\Sigma^{(S_1,S_2)}$ with length $l+1$. By line 25, $z_i(0) = (x_i(0), x_{q,i}(0)) \in Z_{i,0}$. (Step #0) By definition of R_i in line 26 and of $X_0^c(j)$ in line 23, we get

$$\mathbf{d}_i(x_i(0), H_{q,i}(x_{q,i}(0))) \leq \theta_i, \quad i = 1, 2,$$

which, by setting $q_{0,i} = H_{q,i}(x_{q,i}(0))$, implies that the inequality in (13) is satisfied for $t = 0$. By definition of marked states in $T^c(j)$, if P_i blocks at $x_i(0)$ then $x_{q,i}(0)$ is a marked state of T_{Q_i}. As a consequence, $H_{q,i}(x_{q,i}(0)) \in Q_i$ and the statement holds. We now proceed by induction. Suppose P_i does not block at step t.
(Step # t) By definition of h_i^j in line 25 and of $\underset{c,j}{\longrightarrow}$ in line 22, for any $u_i(t) \in h_i^j(z_i(t), x_i(t)) = h_i^j((x_i(t), x_{q,i}(t)), x_i(t))$, for any $x'_{3-i}(t) \in X_{3-i}$, for any $x_i(t+1) \in F_i(x_i(t), x'_{3-i}(t), u_i(t))$, there exist transitions $x_{q,i}(t) \underset{\prime,q,i}{\longrightarrow} x^+_{q,i}(t+1)$ of T'_{Q_i} such that

$$\mathbf{d}_i(x_i(t+1), H_q(x^+_{q,i}(t+1))) \leq \theta_i, \quad i = 1, 2,$$

which, by setting $q_{t+1,i} = H_{q,i}(x_{q,i}(t+1))$, implies that the inequality in (13) is satisfied for step $t+1$. Let $z_i(t+1)$ be a next state of C_i starting from $z_i(t)$ with input $x_i(t)$, i.e. $z_i(t+1) \in G_i(z_i(t), x_i(t))$. By definition of G_i in line 24, of set of states $X^c(j)$ (line 22) and of $\underset{c,j}{\longrightarrow}$ (line 22), we get $(x_1(t+1), x_{q,t+1}(t+1), x_2(t+1), x_{q,2}(t+1)) \in X^c(j)$. Hence, $(x_1(0), x_2(0), z_1(0), z_2(0)) \ldots (x_1(t+1), x_2(t+1), z_1(t+1), z_2(t+1))$ is a state trajectory of $\Sigma^{(S_1,S_2)}$. Analogously to (Step #0), if P_i blocks at step $t+1$ then $H_{q,i}(x_{q,i}(0)) \ldots H_{q,i}(x_{q,i}(t+1)) \in Q_i$ and the statement holds. Suppose now P_i does not block at time $t+1$.

Step # t is repeated until P_i blocks. Since P_i is finite, this procedure converges. Lines 19–23 ensures that transition relation $\underset{c,j}{\longrightarrow}$ respects control architecture, i.e. $u_i(t+1)$ depends only on $x_i(t)$ and $z_i(t)$ but not on $x_{3-i}(t)$. By definition of marked states of $T^c(j)$ and since $T^c(j)$ is co-accessible (line 23), there exists a step $l \in \mathbb{N}$ such that $(x_1(l), x_{q,1}(l), x_2(l), x_{q,2}(l)) \in X^c_m$. As a consequence, $x_{q,i}(l) \in X^m_{q,i}$, $i = 1, 2$. Hence, by setting $q_{l,i} = H_{q,i}(x_{q,i}(l))$, $i = 1, 2$, sequence $q_{0,i}\, q_{1,i} \ldots, q_{l,i} \in \mathcal{L}^y_m(T'_{Q_i}) \subseteq Q_i$, $i = 1, 2$. Hence, the result follows.

Remark 4. It may happen that Algorithm 1 outputs empty strategies for some transition systems T'_{Q_1} and T'_{Q_2} while it outputs non-empty strategies for some sub–transition systems $T''_{Q_1} \sqsubseteq T'_{Q_1}$ and $T''_{Q_2} \sqsubseteq T'_{Q_2}$. This is indeed the case of Example 2.

5.2. Main results

This section provides a characterization of sets $\mathcal{E}$, $\mathcal{N}$ and $\mathcal{D}$.

We now introduce a collection of pairs of strategies that will be used in the sequel to characterize equilibria.

Definition 13. The set $\mathcal{G}$ denotes the finite collection of pairs of strategies (S_1, S_2) that are output of Algorithm 1, for inputs P_i and some inputs $T'_{Q_i} \in \mathrm{Split}(T_{Q_i})$, $i = 1, 2$. For any pair $(S_1, S_2) \in \mathcal{G}$:
- we denote by $\mathcal{G}|_{\mathcal{S}_i}$, the projection of $\mathcal{G}$ onto $\mathcal{S}_i$, i.e. the collection of $S_i \in \mathcal{S}_i$ for which there exists $S_{3-i} \in \mathcal{S}_{3-i}$ such that $(S_1, S_2) \in \mathcal{G}$, $i = 1, 2$;
- $\mathbb{T}_i(S_i, S_{3-i}) = \mathrm{Proj}(T^c(j), T'_{Q_i})$, $i = 1, 2$, where $T^c(j)$ is the transition system from which pair of strategies (S_1, S_2) is derived (see lines 23–26 of Algorithm 1).

Projection operator $\mathrm{Proj}(T^c(j), T'_{Q_i})$ in the definition above is well defined in view of Remark 2. As a direct consequence of Theorem 1, we get $\mathcal{G} \subseteq \mathcal{E}$. Since cardinalities of sets $\mathrm{Split}(T_{Q_1})$ and $\mathrm{Split}(T_{Q_2})$ are finite then set $\mathcal{G}$ is finite, as well. Since cardinality of $\mathcal{E}$ is infinite, $\mathcal{E} \subseteq \mathcal{G}$ is not true. Set $\mathcal{G}$ is partially ordered by $\preceq$ since it is contained in $\mathcal{E}$, and can be viewed as a "generator" of equilibria. In fact, as we show in the subsequent results, all equilibria, Nash and dominant equilibria, can be related to those in $\mathcal{G}$.

We can now give the following result that shows that either a pair of strategies cannot enforce any part of the specifications, or if a pair of strategies can enforce some part of the specification, there also exists another pair that is the output of Algorithm 1 that enforces precisely the same words from the specifications.

Proposition 6. *For any pair of strategies $(S_1, S_2) \in \mathcal{S}_1 \times \mathcal{S}_2$, either one of the following two conditions holds:*
- *$\mathcal{Q}_1(S_1, S_2) = \varnothing$ or $\mathcal{Q}_2(S_2, S_1) = \varnothing$;*
- *there exists a pair $(S'_1, S'_2) \in \mathcal{G}$ such that $\mathcal{Q}_1(S_1, S_2) = \mathcal{Q}_1(S'_1, S'_2)$ and $\mathcal{Q}_2(S_2, S_1) = \mathcal{Q}_2(S'_2, S'_1)$.*

Proof. The first statement holds when either Q_1 or Q_2 cannot be enforced on plants P_1 and P_2, by strategies S_1 and S_2, respectively. Consider the second statement. Consider any $T'_{Q_1} \in \mathrm{Split}(T_{Q_1})$ and $T'_{Q_2} \in \mathrm{Split}(T_{Q_2})$ be such that $\mathcal{Q}_i(S_i, S_{3-i}) \subseteq \mathcal{L}^y_m(T'_{Q_i})$, $i = 1, 2$. Consider transition system $T = (X, X_0, U, \longrightarrow, X_m, Y, H)$ coinciding with transition system T^c of Algorithm 1, except for the transition relation $\longrightarrow$ defined by $(x_1, x_{q,1}, x_2, x_{q,2}) \xrightarrow{(u_1,u_2)} (x^+_1, x^+_{q,1}, x^+_2, x^+_{q,2})$ if there exist a trajectory $(x_1(.), x_2(.), z_1(.), z_2(.))$ of $\Sigma^{(S_1,S_2)}$, a step $t \in \mathbb{N}$ and state runs $r^1 : x_{q,1}(0)x_{q,1}(1) \ldots$ and $r^2 : x_{q,2}(0)x_{q,2}(2) \ldots$ of $T'_{Q,1}$ and $T'_{Q,2}$, respectively, such that

$$(x_1(t), x_{q,1}(t), x_2(t), x_{q,2}(t)) = (x_1, x_{q,1}, x_2, x_{q,2}),$$
$$(x_1(t+1), x_{q,1}(t+1), x_2(t+1), x_{q,2}(t+1)) =$$
$$(x^+_1, x^+_{q,1}, x^+_2, x^+_{q,2}).$$

Transition system T encodes trajectories of $\Sigma^{(S_1,S_2)}$ and state runs of T'_{Q_i}. Let $T' = (X', X'_0, U, \underset{\prime}{\longrightarrow}, X'_m, Y, H') = \mathrm{Trim}(T)$. Since we are considering decentralized control architecture requiring C_i not to depend on x_{3-i}, $i = 1, 2$, transition T' satisfies the following properties:
(i) $T' = \mathrm{Trim}(T')$ is accessible and co-accessible;
(ii) for any $(x_1, x_{q,1}, x_2, x_{q,2}) \in X'$, either $(x_1, x_{q,1}, x_2, x_{q,2}) \in X'_m$ or there exists $(u_1, u_2) \in U_1 \times U_2$ such that $(x_1, x_{q,1}, x'_2, x'_{q,2}) \underset{\prime}{\xrightarrow{(u_1,u_2)}} (x^+_1, x^+_{q,1}, x^+_2, x^+_{q,2})$, for any $(x_1, x_{q,1}, x'_2, x'_{q,2}) \in X'$, and $(x'_1, x'_{q,1}, x_2, x_{q,2}) \underset{\prime}{\xrightarrow{(u_1,u_2)}} (x^+_1, x^+_{q,1}, x^+_2, x^+_{q,2})$, for any $(x'_1, x'_{q,1}, x_2, x_{q,2}) \in X'$.

Let T^c be transition system in line 7 of Algorithm 1 corresponding to P_i and T'_{Q_i}, $i = 1, 2$. Transition system T^c satisfies properties (i) and (ii) above, with $\underset{\prime}{\longrightarrow}$ replaced by $\underset{c}{\longrightarrow}$, from which, $T' \sqsubseteq T^c$. As a consequence, there exists $T'_{Q_1} \in \mathrm{Split}(T_{Q_1})$ and $T'_{Q_2} \in \mathrm{Split}(T_{Q_2})$ and a corresponding $T^c(j) \in \mathrm{Split}(T^c)$ such that the corresponding strategy (S'_1, S'_2) satisfies the statement. □

Define binary relation $\sim$ on $\mathcal{S}_1 \times \mathcal{S}_2$ such that

$$(S_1, S_2) \sim (S'_1, S'_2),$$

if $\mathcal{Q}_1(S_1, S_2) = \mathcal{Q}_1(S'_1, S'_2)$ and $\mathcal{Q}_2(S_2, S_1) = \mathcal{Q}_2(S'_2, S'_1)$. In the sequel we write $(S_1, S_2) \nsim (S'_1, S'_2)$ when either $\mathcal{Q}_1(S_1, S_2) \neq \mathcal{Q}_1(S'_1, S'_2)$ or $\mathcal{Q}_2(S_2, S_1) \neq \mathcal{Q}_2(S'_2, S'_1)$. By definition, $\sim$ is an equivalence relation on $\mathcal{E}$ and also on $\mathcal{S}_1 \times \mathcal{S}_2$. Denote by $\mathcal{E}/_\sim$ and by $\mathcal{S}_1 \times \mathcal{S}_2/_\sim$, the quotients on $\mathcal{E}$ and $(\mathcal{S}_1 \times \mathcal{S}_2)$, respectively, induced by $\sim$. More precisely, given $\mathcal{E}$, let $\mathcal{E}_i, i = 1, 2, \ldots$ be the collection of equivalence classes induced by $\sim$, i.e. such that $(S_1, S_2), (S'_1, S'_2) \in \mathcal{E}_i$ if and only if $(S_1, S_2) \sim (S'_1, S'_2)$, and such that $\mathcal{E} = \cup_i \mathcal{E}_i$. By construction, $\mathcal{E}_i \cap \mathcal{E}_j = \varnothing$, if $i \neq j$. The quotient $\mathcal{E}/_\sim$ of $\mathcal{E}$ induced by $\sim$ is then defined as a collection of strategies $(S_1, S_2) \in \mathcal{E}$, each one being one representative of the equivalence class $\mathcal{E}_i$. Note that $\mathcal{E}/_\sim$ is not unique. The quotient

$\mathcal{S}_1 \times \mathcal{S}_2/_\sim$, on $(\mathcal{S}_1 \times \mathcal{S}_2)$ induced by $\sim$, can be defined, by following a similar reasoning. For later purposes, define $S_i^\varnothing = (C_i, \varnothing)$, for some controllers C_i, $i = 1, 2$, which implies $\mathcal{Q}_1(S_1^\varnothing, S_2^\varnothing) = \varnothing$ or $\mathcal{Q}_2(S_2^\varnothing, S_1^\varnothing) = \varnothing$. We call the pair $(S_1^\varnothing, S_2^\varnothing)$ the null strategy because $\Sigma^{(S_1^\varnothing, S_2^\varnothing)}$ has no trajectories.

The following result shows that pairs of strategies in $\mathcal{E}/_\sim$ and in $(\mathcal{S}_1 \times \mathcal{S}_2)/_\sim$ are equivalent, in the sense of the equivalence relation $\sim$, to pairs of strategies in $\mathcal{G}$ and $\mathcal{G} \cup \{(S_1^\varnothing, S_2^\varnothing)\}$, and vice versa, respectively. This is important to provide a characterization of equilibria, as detailed in the subsequent results.

Theorem 2. *The following statements hold:*
(i) For any pair of strategies $(S_1, S_2) \in \mathcal{E}/_\sim$ there exists a pair of strategies $(S_1', S_2') \in \mathcal{G}$ such that $(S_1, S_2) \sim (S_1', S_2')$, and vice versa;
(ii) For any pair of strategies $(S_1, S_2) \in (\mathcal{S}_1 \times \mathcal{S}_2)/_\sim$ there exists a pair of strategies $(S_1', S_2') \in \mathcal{G} \cup \{(S_1^\varnothing, S_2^\varnothing)\}$ such that $(S_1, S_2) \sim (S_1', S_2')$, and vice versa.

Proof. (Proof of (i)). By Proposition 6, for any $(S_1, S_2) \in \mathcal{E}$ there exists $(S_1', S_2') \in \mathcal{G}$ such that $(S_1, S_2) \sim (S_1', S_2')$. Consider any pair $(S_1', S_2'), (S_1'', S_2'') \in \mathcal{G}$. By definition of $\mathcal{G}$, if $(S_1', S_2') \neq (S_1'', S_2'')$ then $\mathbb{T}_1(S_1, S_2) \neq \mathbb{T}_1(S_1', S_2')$ and $\mathbb{T}_2(S_2, S_1) \neq \mathbb{T}_2(S_2', S_1')$. Consequently, $(S_1', S_2') \nsim (S_1'', S_2'')$. Since $\mathcal{G}$ is the set of minimal cardinality containing a representative of each equivalence class of the quotient, the result follows.
(Proof of (ii)). We get

$$\begin{aligned}(\mathcal{S}_1 \times \mathcal{S}_2)/_\sim &= (\mathcal{E} \cup ((\mathcal{S}_1 \times \mathcal{S}_2)\backslash\mathcal{E}))/_\sim \\ &= (\mathcal{E}/_\sim) \cup (((\mathcal{S}_1 \times \mathcal{S}_2)\backslash\mathcal{E})/_\sim) \\ &= \mathcal{G} \cup (((\mathcal{S}_1 \times \mathcal{S}_2)\backslash\mathcal{E})/_\sim),\end{aligned} \tag{18}$$

where second equality holds because the quotient of the union of disjoint sets is the union of the quotients of these sets, and last equality holds by the proof of statement (i). By definition of $\mathcal{E}$, for any $(S_1, S_2) \in ((\mathcal{S}_1 \times \mathcal{S}_2)\backslash\mathcal{E})$, we get $\mathcal{Q}_1(S_1, S_2) = \varnothing$ or $\mathcal{Q}_2(S_2, S_1) = \varnothing$ from which, $(S_1, S_2) \sim (S_1^\varnothing, S_2^\varnothing)$, implying in turn, $(((\mathcal{S}_1 \times \mathcal{S}_2)\backslash\mathcal{E})/_\sim) = \{(S_1^\varnothing, S_2^\varnothing)\}$. The equality above, combined with (18), implies the statement. □

We now have all the ingredients to present the characterization of sets $\mathcal{E}$, $\mathcal{N}$ and $\mathcal{D}$.

Corollary 1. *Pair $(S_1, S_2) \in \mathcal{S}_1 \times \mathcal{S}_2$ is an equilibrium if and only if there exists $(S_1', S_2') \in \mathcal{G}$ such that $(S_1, S_2) \sim (S_1', S_2')$.*

Corollary 2. *An equilibrium $(S_1, S_2) \in \mathcal{E}$ is Nash if and only if there exists $(S_1', S_2') \in \mathcal{G}$ such that $(S_1, S_2) \sim (S_1', S_2')$ and (S_1', S_2') is Nash in $\mathcal{G}$, i.e.*
- $\mathcal{Q}_1(S_1'', S_2') \subseteq \mathcal{Q}_1(S_1', S_2')$, $\forall S_1'' \in \mathcal{G}|_{\mathcal{S}_1}$,
- $\mathcal{Q}_2(S_1', S_2'') \subseteq \mathcal{Q}_2(S_1', S_2')$, $\forall S_2'' \in \mathcal{G}|_{\mathcal{S}_2}$.

Corollary 3. *An equilibrium (S_1, S_2) is dominant if and only if there exists $(S_1', S_2') \in \mathcal{G}$ such that $(S_1, S_2) \sim (S_1', S_2')$ and (S_1', S_2') is dominant in $\mathcal{G}$, i.e.*
- $\mathcal{Q}_1(S_1'', S_2'') \subseteq \mathcal{Q}_1(S_1', S_2')$, $\forall S_1'' \in \mathcal{G}|_{\mathcal{S}_1}$, $\forall S_2'' \in \mathcal{G}|_{\mathcal{S}_2}$;
- $\mathcal{Q}_2(S_2'', S_1'') \subseteq \mathcal{Q}_2(S_2', S_1')$, $\forall S_2'' \in \mathcal{G}|_{\mathcal{S}_2}$, $\forall S_1'' \in \mathcal{G}|_{\mathcal{S}_1}$.

For the meaning of $\mathcal{G}|_{\mathcal{S}_i}$ we refer to Definition 13. The proof of Corollary 1 is a direct consequence of Theorem 2, while the proofs of Corollaries 2 and 3 are direct consequences of Theorem 2 and Proposition 6. Verifying conditions of Corollaries 1, 2 and 3 requires tools for checking equivalence and inclusions of regular languages well studied in the literature, see e.g. Cassandras and Lafortune (1999) and the references therein.

We now show equivalent formulations of Corollaries 2 and 3 which simply require checking whether a transition system is a sub-transition system of another one. We first need the following technical result.

Lemma 1. *Let $(S_1', S_2') \in \mathcal{G}$ (resp. $(S_1'', S_2'') \in \mathcal{G}$) be the output of Algorithm 1 with inputs P_1 and P_2, and T_{Q_i}', $i = 1, 2$, (resp. T_{Q_i}'', $i = 1, 2$). Then $(S_1', S_2') \preceq (S_1'', S_2'')$ if and only if $T_{Q_1}' \sqsubseteq T_{Q_1}''$ and $T_{Q_2}' \sqsubseteq T_{Q_2}''$.*

Proof. Sufficiency is a direct consequence of Definition 9 and necessity is a direct consequence of Proposition 2. □

The following pair of results show that for assessing if an equilibrium is Nash or dominant, it is sufficient to look at the finite collection of sub-transition systems related to $\mathcal{G}$. These results show then that the problem of finding Nash and dominant equilibria is decidable.

Theorem 3. *An equilibrium $(S_1, S_2) \in \mathcal{E}$ is Nash if and only if there exists $(S_1', S_2') \in \mathcal{G}$ such that $(S_1, S_2) \sim (S_1', S_2')$ and the following conditions hold*
- $\mathbb{T}_1(S_1'', S_2') \sqsubseteq \mathbb{T}_1(S_1', S_2')$, $\forall S_1'' \in \mathcal{G}|_{\mathcal{S}_1}$;
- $\mathbb{T}_2(S_2'', S_1') \sqsubseteq \mathbb{T}_2(S_2', S_1')$, $\forall S_2'' \in \mathcal{G}|_{\mathcal{S}_2}$.

Proof. Direct consequence of Lemma 1 and of Corollary 2. □

Theorem 4. *An equilibrium (S_1, S_2) is dominant if and only if there exists $(S_1', S_2') \in \mathcal{G}$ such that $(S_1, S_2) \sim (S_1', S_2')$ and the following conditions hold*
- $\mathbb{T}_1(S_1'', S_2'') \sqsubseteq \mathbb{T}_1(S_1', S_2')$, $\forall S_1'' \in \mathcal{G}|_{\mathcal{S}_1}$, $\forall S_2'' \in \mathcal{G}|_{\mathcal{S}_2}$;
- $\mathbb{T}_2(S_2'', S_1') \sqsubseteq \mathbb{T}_2(S_2', S_1')$, $\forall S_2'' \in \mathcal{G}|_{\mathcal{S}_2}$, $\forall S_1'' \in \mathcal{G}|_{\mathcal{S}_1}$.

Proof. Direct consequence of Lemma 1 and of Corollary 3. □

We conclude this section with the following

Remark 5. Generally speaking, if a dominant equilibrium does not exist, one is interested in finding a Nash equilibrium. To this purpose, one can use a Hasse diagram to represent the poset $(\text{Split}(T_{Q_1}) \times \text{Split}(T_{Q_2}), \sqsubseteq')$. A Hasse diagram associated with $(\text{Split}(T_{Q_1}) \times \text{Split}(T_{Q_2}), \sqsubseteq')$ is a directed acyclic graph $H = (V, E)$, where $V = \text{Split}(T_{Q_1}) \times \text{Split}(T_{Q_2})$ and $((T_1, T_2), (T_1', T_2')) \in E$, if $(T_1, T_2) \sqsubseteq' (T_1', T_2')$ or equivalently, $T_1 \sqsubseteq T_1'$ and $T_2 \sqsubseteq T_2'$. Then, one may start by applying Algorithm 1 to sinks of H, so that maximal elements of the poset are visited, and it is possible to verify whether they correspond to Nash equilibria. If Nash equilibria are not found, then one can proceed backwards by considering vertices of H pointing to the sinks, and so on, until the roots of H are eventually visited. Moreover, there is no need to construct the full H from the early beginning. Indeed, by using on-the-fly based approach (see e.g. Borri et al. (2012), Gerth et al. (1996) and Pola et al. (2012)), one can start by defining the sinks of H and by looking for Nash equilibria. If no solution is found, one can construct other vertices of H pointing to the sinks. This can be done by removing one state at a time from T_{Q_1} and T_{Q_2} and by applying the Trim operator to the transition systems obtained. Then, one can check existence of Nash equilibria for these new vertices of H. If no solution is found, this procedure is iterated. This approach may reduce in some cases time computational complexity.

6. An illustrative example

In this section we consider an example in the context of multi-agent systems with shared resources. Agents P_1 and P_2 move on a rectangle $X_1 = X_2 = [1; 6] \times [1; 4]$. Fig. 6 depicts the set of states of P_1 and P_2, together with their initial states $x_{1,0}$ and $x_{2,0}$, and target states $\alpha, \beta, \gamma, \delta$. Inputs of P_i are "North" (n), "South" (s), "East" (e) or "West" (w), where $n = (0, 1)$, $s = (0, -1)$, $e = (-1, 0)$ and $w = (1, 0)$. So, if P_1 is in state (3, 3) and input

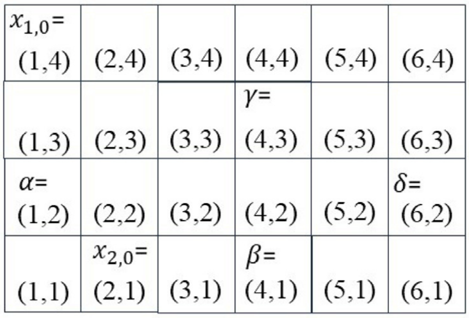

Fig. 6. Example of Section 6.

is w, its next state is $(3,3)+(1,0)=(4,3)$. If P_1 is in $(1,1)$ and input is s, it blocks because $(1,1)+(0,-1)=(1,0)\notin X_1=X_2$. Dynamics of agents P_i are given by (7), where $X_i=[1;6]\times[1;4]$, $X_{1,0}=\{(1,4)\}$, $X_{2,0}=\{(2,1)\}$, $U_i=\{n,s,e,w\}$, and F_i is described by

$$F_i(x_i,x_{3-i},u_i)=(\{x_i+u_i\}\backslash(\{x_{3-i}\}+U_{3-i}))\cap X_i.$$

The dynamics above prevent collisions between P_1 and P_2 from happening. Indeed, if P_1 is in x_1 and P_2 is in x_2, next states of P_1 and P_2 are different, independently from the choices of u_1 and u_2. However, it may happen that P_i blocks, which corresponds to the situation where there is no action that ensures no collision and being in $X_1=X_2$. For example, if P_1 is in $(1,1)$ and P_2 in $(2,2)$, P_1 blocks. Indeed, if P_1 picks inputs n or w then P_2 can pick inputs e or s respectively, leading to a collision; if P_1 picks inputs e or s, it goes outside $X_1=X_2$.

The goal of each P_i is to visit target states α, β, γ and δ. More precisely, the specifications considered are languages defined over the alphabet $X_1=X_2$, and described by:

$$Q_1=Q_{\{\beta,\gamma\},1}\cup Q_\alpha,\quad Q_2=Q_{\{\beta,\gamma\},2}\cup Q_\delta, \tag{19}$$

where:

- $Q_{\{\beta,\gamma\},i}$: agent P_i is required to reach set $\{\beta,\gamma\}$ and to alternate visits of β and γ (not necessarily in this order) an arbitrarily large number of times;
- Q_α : P_1 is required to reach α and to possibly stay in α an arbitrarily large number of times;
- Q_δ : P_2 is required to reach δ and to possibly stay in δ an arbitrarily large number of times.

Specifications above can be formalized as follows:

$$Q_{\{\beta,\gamma\},i}=Q_{\beta\gamma,i}\cup Q_{\gamma\beta,i},\ i=1,2,$$
$$Q_{\beta\gamma,i}=\{x_{i,0}\}\,X_i^*(\{\beta\}\,X_i^*\,\{\gamma\})(\{\beta\}\,X_i^*\,\{\gamma\})^*,\ i=1,2,$$
$$Q_{\gamma\beta,i}=\{x_{i,0}\}\,X_i^*(\{\gamma\}\,X_i^*\,\{\beta\})(\{\gamma\}\,X_i^*\,\{\beta\})^*,\ i=1,2,$$
$$Q_\alpha=\{x_{1,0}\}\,X_1^*\{\alpha\}\{\alpha\}^*,$$
$$Q_\delta=\{x_{2,0}\}\,X_2^*\{\delta\}\{\delta\}^*.$$

Since $\{x_{i,0}\}$, X_i, $\{\alpha\}$, $\{\beta\}$, $\{\gamma\}$ and $\{\delta\}$ are regular languages, by Proposition 1, languages Q_1 and Q_2 in (19) are regular. Transition systems T_{Q_1} and T_{Q_2}, marking Q_1 and Q_2, are depicted in Figs. 7 and 8, respectively. Each circle represents a state; upper label is the name of the state, lower label is its output. Note that lower label in states 2, 6, 7, 11, 12, 15, 16 is X_1, and lower label in states $2'$, $6'$, $7'$, $11'$, $12'$, $15'$, $16'$ is X_2. This is a short hand notation for not representing all the 48 states of X_1 and X_2 and all possible transitions between all of them. Since inputs in T_{Q_1} and T_{Q_1} play no role, labels on their arrows are not depicted, for simplicity.

By applying the results presented in the previous sections with $\theta_1=\theta_2=0$, we compute $\mathcal{G}$ as detailed in Table 1. The

Table 1
Set $\mathcal{G}$ of equilibria of the example of Section 6 in the first column. Corresponding transition systems used as input of Algorithm 1 to generate such equilibria, in the second column. Corresponding parts of the specifications enforced, in the third and forth columns.

$\mathcal{G}$	$\text{Split}(T_{Q_1})\times$ $\text{Split}(T_{Q_2})$	$\mathcal{Q}_1(\cdot)$	$\mathcal{Q}_2(\cdot)$
$E_1=(S_{1,\beta\gamma},S_{2,\gamma\beta})$	$(T_{Q_1}^{E_1},T_{Q_2}^{E_1})$	$Q_{\beta\gamma}$	$Q_{\gamma\beta}$
$E_2=(S_{1,\gamma\beta},S_{2,\beta\gamma})$	$(T_{Q_1}^{E_2},T_{Q_2}^{E_2})$	$Q_{\gamma\beta}$	$Q_{\beta\gamma}$
$E_3=(S_{1,\alpha},S_{2,\beta\gamma})$	$(T_{Q_1}^{E_3},T_{Q_2}^{E_3})$	Q_α	$Q_{\beta\gamma}$
$E_4=(S_{1,\alpha},S_{2,\gamma\beta})$	$(T_{Q_1}^{E_4},T_{Q_2}^{E_4})$	Q_α	$Q_{\gamma\beta}$
$E_5=(S_{1,\beta\gamma+\alpha},S_{2,\beta\gamma})$	$(T_{Q_1}^{E_5},T_{Q_2}^{E_5})$	$Q_{\beta\gamma}\cup Q_\alpha$	$Q_{\beta\gamma}$
$E_6=(S_{1,\gamma\beta+\alpha},S_{2,\gamma\beta})$	$(T_{Q_1}^{E_6},T_{Q_2}^{E_6})$	$Q_{\gamma\beta}\cup Q_\alpha$	$Q_{\gamma\beta}$

second column details the sub-transition systems $(T'_{Q_1},T'_{Q_2})\in \text{Split}(T_{Q_1})\times\text{Split}(T_{Q_2})$ from which equilibria are obtained. By setting $T_{Q_i}^{E_j}=T_{Q_i}|_{X^{(i,j)}}$, for $i=1,2$, $j=1,2,\ldots,6$ (notation $T_{Q_i}|_{X^{(i,j)}}$ corresponds to the notion of sub-transition system induced by a set, see Definition 3), we get:

$$X^{(1,1)}=\{1,2,3,6,9,11,13,15\},$$
$$X^{(1,2)}=\{1,2,4,7,10,12,14,16\},$$
$$X^{(1,3)}=X^{(1,4)}=\{1,2,5,8\},$$
$$X^{(1,5)}=X^{(1,1)}\cup X^{(1,3)},$$
$$X^{(1,6)}=X^{(1,2)}\cup X^{(1,3)},$$
$$X^{(2,1)}=X^{(2,4)}=X^{(2,6)}=\{1',2',4',7',10',12',14',16'\},$$
$$X^{(2,2)}=X^{(2,3)}=X^{(2,5)}=\{1',2',3',6',9',11',13',15'\}.$$

Controllers of strategies

$$S_{i,W}=(C_{i,W},\{x_{i,0}\}) \tag{20}$$

in Table 1, with $i=1,2$ and $W=\beta\gamma,\gamma\beta,\alpha$ are detailed below

$$C_{1,\beta\gamma}(x_{1,0})=\{(s\,s\,s\,r\,r\,r)\}\{(r\,n\,n\,l\,l\,s\,s\,r)\}^*,$$
$$C_{1,\gamma\beta}(x_{1,0})=\{(r\,r\,r\,r\,l\,s)\}\{(l\,s\,s\,r\,r\,n\,n\,l)\}^*,$$
$$C_{1,\alpha}(x_{1,0})=\{(s\,n\,s\,n\,s\,s)\}\{(n\,n\,s\,n\,s\,n\,s\,s)\}^*,$$
$$C_{2,\gamma\beta}(x_{2,0})=\{(r\,r\,r\,n\,n\,l)\}\{(l\,s\,s\,r\,r\,n\,n\,l)\}^*,$$
$$C_{2,\beta\gamma}(x_{2,0})=\{(l\,r\,l\,r\,r\,r)\}\{(r\,n\,n\,l\,l\,s\,s\,r)\}^*,$$

where languages above list sequences of controls to be applied. Equilibria E_5 and E_6 can be viewed as the "union" of equilibria E_1 and E_3, and of E_1 and E_4, respectively. Corresponding strategies are given by:

$$S_{1,\beta\gamma+\alpha}=(C_{1,\beta\gamma+\alpha},\{x_{1,0}\}),$$
$$S_{1,\gamma\beta+\alpha}=(C_{1,\gamma\beta+\alpha},\{x_{1,0}\}),$$
$$C_{1,\beta\gamma+\alpha}(x_{1,0})=C_{1,\beta\gamma}(x_{1,0})\cup C_{1,\alpha}(x_{1,0}),$$
$$C_{1,\gamma\beta+\alpha}(x_{1,0})=C_{1,\gamma\beta}(x_{1,0})\cup C_{1,\alpha}(x_{1,0}).$$

Strategy $S_{1,\beta\gamma+\alpha}$ defined above reads as follows. Starting from $x_{1,0}$, controller $C_{1,\beta\gamma+\alpha}$ enforces control sequences $(s\,s\,s\,r\,r\,r)$ or $(s\,n\,s\,n\,s\,s)$. By picking the first (resp. second) control sequence, state β (resp. α) is reached at step 6. When in state β (resp. α), control sequence chosen is $(r\,n\,n\,l\,l\,s\,s\,r)^*$ (resp. $(n\,n\,s\,n\,s\,n\,s\,s)^*$) by which, state γ (resp. α) is reached at step 10 and then state β (resp. α) is reached again at step 14. Hence, specification enforced by $S_{1,\beta\gamma}$ on P_1 is $Q_{\beta\gamma}$. Same reasoning applies to strategies $S_{1,\gamma\beta+\alpha}$ and in (20).

It is readily seen that all strategies detailed above are not unique and that, by this choice of strategies, corresponding agents' dynamics do not block.

Some comments are in order. For P_2 to reach state δ, an odd number of steps is needed. For P_1 and P_2 to reach states α, β

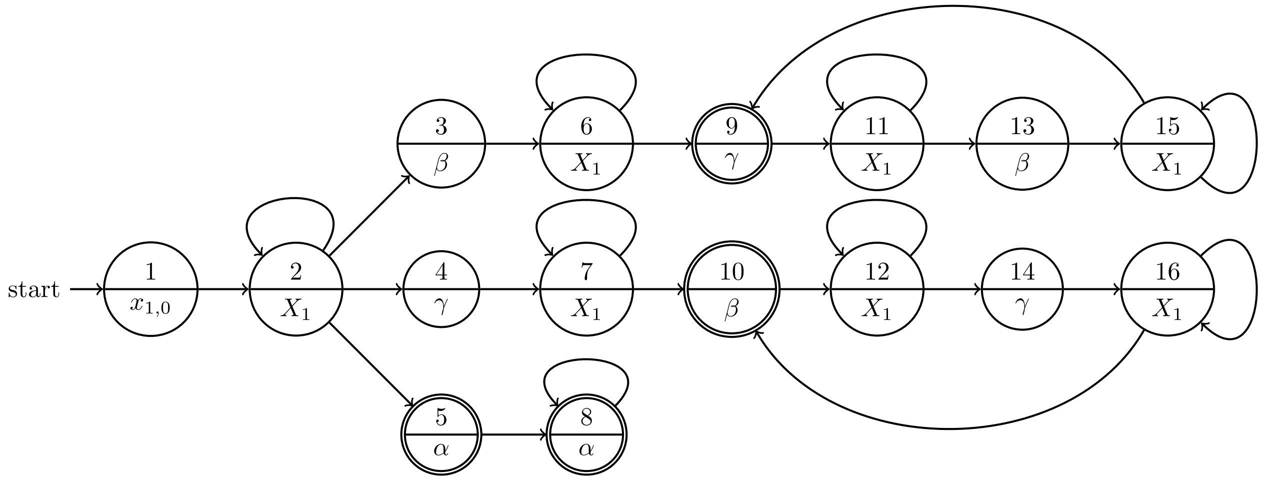

Fig. 7. Transition system T_{Q_1} in the example of Section 6.

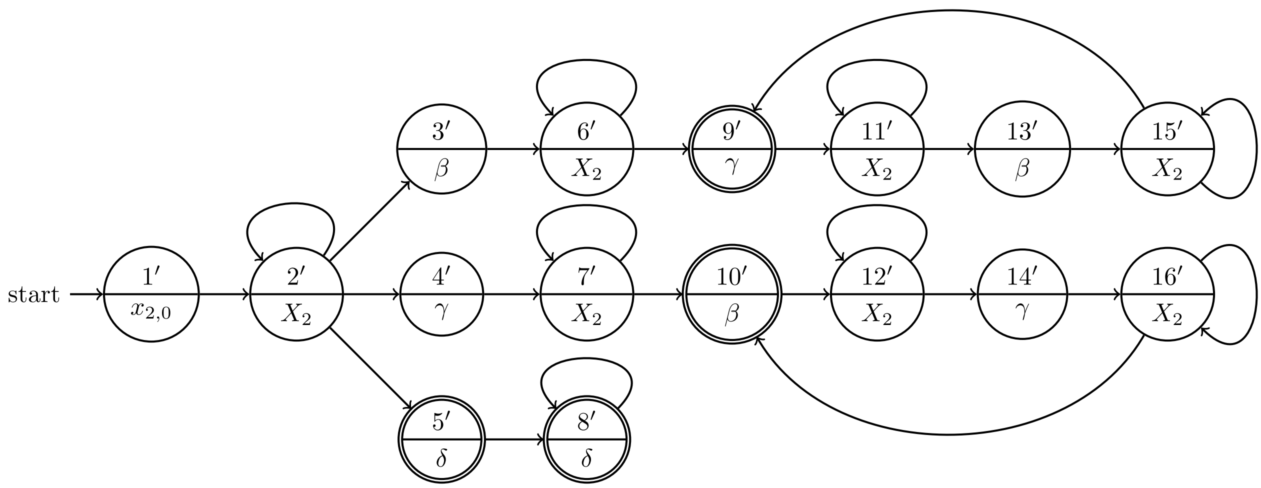

Fig. 8. Transition system T_{Q_2} in the example of Section 6.

and γ an even number of steps is instead needed. This lack of "synchronism" does not allow specification Q_δ to be enforced. By looking at Table 1, while E_5 and E_6, obtained as the "union" of E_1 and E_3 and, of E_1 and E_4, respectively, are in $\mathcal{G}$, the "union" of equilibria E_1 and E_2 is not. This is because if P_1 chooses actions in strategy $S_{1,\beta\gamma}$ and P_2 chooses actions in strategy $S_{2,\beta\gamma}$, ending states would coincide in β, and this collision is avoided by definition of the plants' dynamics.

By Table 1 (last two columns), resulting Nash equilibria in $\mathcal{G}$ are:

$$\mathcal{N} \cap \mathcal{G} = \{E_5, E_6\},$$

and none of them is dominant. In this example, dynamics of P_i are deterministic. The model could be enriched by considering external inputs that are neither controllable nor measurable, leading to a nondeterministic system. This more general situation can be addressed with the results shown, but is not illustrated here for notational simplicity.

7. Conclusions

In this paper, we proposed the notions of equilibria, Nash equilibria and dominant equilibria for investigating decentralized control problems of interconnected nondeterministic and metric finite state systems with regular language specifications. We showed that characterizing these notions is a decidable problem. An application to multi-agent systems with shared resources was also presented. Software implementation of the proposed results is a direction for future investigation.

Acknowledgment

The authors wish to thank the anonymous reviewers for their very useful comments and suggestions.

References

Alur, R., Henzinger, T., Kupferman, O., & Vardi, M. (1998). Alternating refinement relations. In *Lecture notes in computer science*, *Proceedings of the 8th international conference on concurrence theory* (pp. 163–178). Springer, number 1466.

Belta, C., Yordanov, B., & Gol, E. Aydin (2017). *Formal methods for discrete-time dynamical systems*. Springer, 2017.

Borri, A., Pola, G., & Di Benedetto, M. D. (2012). Integrated symbolic design of unstable nonlinear networked control systems. In *51th IEEE conference on decision and control* (pp. 1374–1379).

Cassandras, C. G., & Lafortune, S. (1999). *Introduction to discrete event systems*. Kluwer Academic Publishers.

Caugherty, Dan (1990). JFLAP: An interactive formal languages and automata package. Available online at http://www.jflap.org/.

Clarke, E. M., Grumberg, O., & Peled, D. (1999). *Model checking*. MIT Press.

Gerth, R., Peled, D., Vardi, M. Y., & Wolper, P. (1996). Simple on-the-fly automatic verification of linear temporal logic. In Piotr Dembinski, & Marek Sredniawa (Eds.), *IFIP conference proceedings*, *Protocol specification, testing and verification XV, proceedings of the 15th IFIP workshop* (pp. 3–18). Warsaw: Chapman & Hall.

Girard, A., & Pappas, G. J. (2007). Approximation metrics for discrete and continuous systems. *IEEE Transactions on Automatic Control*, *52*(5), 782–798.

Gol, E. A., Lazar, M., & Belta, C. (2014). Language–guided controller synthesis for linear systems. *IEEE Transactions of Automatic Control*, *59*(5), 1163–1176.

Jiang, S., Kumar, R., Takai, S., & Qiu, W. (2010). Decentralized control of discrete-event systems with multiple local specifications. *IEEE Transactions on Automation Science and Engineering*, *7*(3), 512–522.

Lawson, M. V. (2004). *Finite automata*. CRC Press.

Masciulli, T., Pola, G., De Santis, E., & Di Benedetto, M. D. (2021). Output feedback reachability of controlled-observable states for nondeterministic finite-state systems. *IEEE Control Systems Letters*, *6*, 464–469.

Pola, G., Borri, A., & Di Benedetto, M. D. (2012). Integrated design of symbolic controllers for nonlinear systems. *IEEE Transactions on Automatic Control*, *57*(2), 534–539.

Pola, G., De Santis, E., & Di Benedetto, M. D. (2023). On Nash equilibria for decentralized symbolic control of interconnected finite state systems. In *62nd IEEE conference on decision and control*. Singapore.

Pola, G., De Santis, E., & Di Benedetto, M. D. (2024). Decentralized control of networks of nondeterministic and metric finite state systems. *IEEE Control Systems Letters*, *8*, 1463–1468.

Pola, G., & Di Benedetto, M. D. (2019). Control of cyber-physical-systems with logic specifications: A formal methods approach. *Annual Reviews in Control*, *47*, 178–192.

Pola, G., Pepe, P., & Di Benedetto, M. D. (2016). Symbolic models for networks of control systems. *IEEE Transactions on Automatic Control*, *61*(11), 3663–3668.

Pola, G., Pepe, P., & Di Benedetto, M. D. (2018). Decentralized approximate supervisory control of networks of nonlinear control systems. *IEEE Transactions on Automatic Control*, *63*, 2803–2817.

Pola, G., & Tabuada, P. (2009). Symbolic models for nonlinear control systems: Alternating approximate bisimulations. *SIAM Journal on Control and Optimization*, *48*(2), 719–733.

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

Tabuada, P. (2008). An approximate simulation approach to symbolic control. *IEEE Transactions on Automatic Control*, *53*(6), 1406–1418.

Tabuada, P. (2009). *Verification and control of hybrid systems: A symbolic approach*. Springer.

Giordano Pola received the "Laurea degree" in electrical engineering and the Ph.D. degree in electrical engineering and computer science from the University of L'Aquila, Italy, in July 2000 and June 2004, respectively. He was a Visiting Scholar with the University of Cambridge, U.K., in 2002 and 2003, and an Assistant Researcher in 2003 and a Research Fellow in 2004 with the University of Twente, The Netherlands. From October 2006 to December 2007, he was a Postdoctoral Researcher with the University of California at Los Angeles, USA. Since 2018 he has been an Associate Professor with the University of L'Aquila. His research interests include modeling, analysis, and control of hybrid, embedded, networked, and distributed systems.

Dr. Pola was the recipient of Fondazione Filauro Award for Ph.D. students in 2003. He was a Plenary Speaker at the First International Conference on Systems and Computer Science, 2012. He is currently Chair of the Master Degree in Control Systems and Automation Engineering and member of the Executive Board of the Center of Excellence for Research DEWS at the University of L'Aquila. Since 2015, he has been a Senior Member of IEEE.He served as Associate Editor of International Journal of Control (2019-2023) and he is currently serving as Associate Editor of IEEE Transactions on Automatic Control.

Elena De Santis is a Full Professor in Systems and Control Engineering with the Department of Information Engineering, Computer Science and Mathematics (DISIM), University of L'Aquila Since 2019, she has been the Director of the Italian Center of Excellence for Research DEWS "Architectures and Design methodologies for Embedded controllers, Wireless interconnect and System-on-Chip". She graduated (summa cum laude) in Electrical Engineering from the University of L'Aquila in 1983 and joined the same university as an assistant professor in 1987. Since then, she has published more than 100 papers on analysis, control and optimization of constrained and uncertain dynamic systems, model management for decision support systems, positive systems, dynamic games, hybrid and switching systems. She has been a member of the IEEE since 1999 (elevated to the rank of Senior Member in 2006) and of the IFAC TC 1.3 Committee on Discrete Event and Hybrid Systems. She has served as a member of numerous conference program committees and has been a member of the editorial board of the European Journal of Control and Nonlinear Analysis, Hybrid Systems. She was a guest editor of the Special Issue: "Observability and observer-based control of hybrid systems", Int. J. of Robust and Nonlinear Control, 19(4), 2009, and coauthor (with M.D. Di Benedetto) of the monographs *Observability of Hybrid Dynamical Systems*, NOW 2016, and *H-systems: Observability, Diagnosability, and Predictability of Hybrid Dynamical Systems*, Springer 2023.

Maria Domenica Di Benedetto is Professor Emeritus at University of L'Aquila (Italy). She received her Master degree (summa cum laude) in Electrical Engineering and Computer Science from University of Roma "La Sapienza", and the "Docteur-Ingénieur" and "Doctorat d'Etat ès Sciences" degrees from the Université de Paris-Sud (Orsay, France). She has been a Professor of Automatic Control at University of L'Aquila and Adjunct Professor and McKay Professor at the University of California at Berkeley. She held visiting positions at MIT, at University of Michigan Ann Arbor, and at Ecole Nationale Supérieure de Mécanique in Nantes (France).

She is an IFAC Fellow and a Life Fellow of the IEEE. Her research interests are in the areas of nonlinear and hybrid systems control theory, diagnosability and predictability in cyber–physical systems, and applications to traffic control, smart grids and biological systems.

From 2001 to 2019, Professor Di Benedetto has been the PI and Director of the Italian Center of Excellence for Research DEWS. She has been the President of the Italian Association of Researchers in Automatic Control (SIDRA) from 2013 to 2019. She is the President of the European Embedded Control Institute since 2009. She is Editor of the IEEE Press Book Series in Control Systems Theory and Applications.