

Distributed games and distributed control for asynchronous systems

Paul Gastin*, Benjamin Lerman*, and Marc Zeitoun*

LIAFA, Université Paris 7 & CNRS
2, pl. Jussieu, case 7014
F-75251 Paris cedex 05, France

Paul.Gastin, Benjamin.Lerman, Marc.Zeitoun@liafa.jussieu.fr

Abstract. We introduce distributed games over *asynchronous* transition systems to model a distributed controller synthesis problem. A game involves two teams and is not turn-based: several players of both teams may simultaneously be enabled. We define distributed strategies based on the *causal* view that players have of the system. We reduce the problem of finding a winning distributed strategy with a given memory to finding a memoryless winning distributed strategy in a larger distributed game. We reduce the latter problem to finding a strategy in a classical 2-players game. This allows to transfer results from the sequential case to this distributed setting.

Keywords. Distributed game, distributed control, distributed strategy.

1 Introduction

The controller synthesis problem has been widely investigated by many authors for different system types (sequential, concurrent, timed, probabilistic) and different specification languages (linear or branching temporal logics for instance). The variant addressed here, called distributed controller synthesis problem, is the following. We are given a *distributed reactive system* executing a program in some environment, modeled by an *asynchronous* transition system [14] made up of several processes. It can perform local actions and synchronization actions involving several processes. Such an action first reads states of the participating processes and, depending on what is read, chooses a transition changing the states of the involved processes. Interpreting processes as memory locations, this suggests communication via shared memory, and explains the terminology adopted in the paper. However, one can as well simulate with these asynchronous systems other communication paradigms, such as point-to-point channels. Another advantage of this model is to handle actions of the environment just as actions of the reactive system. We are also given a *specification*, *i.e.*, a property expressing behaviors one wants to ensure for the system. The distributed controller synthesis problem is then to compute a distributed controller on the

* Work partly supported by the European research project HPRN-CT-2002-00283 GAMES and by the ACI Sécurité Informatique 2003-22 (VERSYDIS).

same process set (actions of the controller observe local states of some processes). With this information, the controller has to enable or disable controllable actions of the system, so that the *overall* system behaves correctly according to the specification.

The problem is known to be decidable and to have an optimal solution in the sequential case [11], *i.e.*, when there is a single process. For distributed systems however, the situation is more involved. Several models have been considered so far (formulated in control or in game theory terminology). For synchronous processes communicating via buffers, the problem is undecidable [10] for LTL specifications except for very few communication architectures. Recent works [5,4] extend this result, *e.g.*, for local specifications (talking only of actions of single processes). The approach of [7] unifies [10,5,4]. In all settings, a major reason for undecidability is when the specification language makes it possible to express properties of an *observed linearization* of process actions, ignoring their possible concurrency. Another distributed model is studied in [6], and it is shown that the existence of specific controllers is decidable for specification languages making no distinction between linearizations of the same concurrent execution.

Systems used in this paper subsume those of [10,5,4,7] and [6]. In [6], global transitions of the system are obtained by synchronizing local transitions of processes, and transitions of the environment are local. Here in contrast, a transition of a synchronizing action also depends on the states of other involved processes, so that transition functions of actions are not necessarily a cartesian product of local transition functions. Further, environment moves can be defined globally. Systems of [7], in which the environment is global and transitions of the system are purely local (process communication flows through the environment) can also be modeled naturally in our framework. Another difference is that [6,7] use *local* memory controllers, based on the history of process local states (cf. Sec. 5). We use *causal* memory: a controller can remember information collected from other processes along the computation. The existence of a distributed controller in the settings of [6,7] implies the existence of a controller in our setting. As the converse does not hold, one cannot transfer immediately the undecidability results of [6,7] to our case.

Our primary goal is to model the distributed controller synthesis problem by *games*. Sequential 2-players games already provide a natural and widely used context to model sequential reactive systems [9,12,15,13]. Player 0 represents the system and player 1 represents the environment. The rules of the game describe the possible interactions between them, and the winning condition for player 0 expresses the specification that the system should meet. Thus, deciding whether player 0 has a winning strategy corresponds to deciding whether the system can be controlled to meet the specification, and computing a winning strategy for player 0 corresponds to solving the controller synthesis problem.

Distributed games proposed in this paper fit suitably to the model of asynchronous systems and supply a natural framework for studying the distributed controller problem. Two teams play one against the other. Players of team 0 may be viewed as controllable actions of a distributed system which cooperate in or-

der to meet the specification, no matter how the environment (team 1) behaves. All players use a pool of shared variables to transmit information. The game is not turn-based: in each position of the game, several players of team 0 or team 1 may be simultaneously enabled. Thus, the game is asynchronous, contrary to the setting of [7] where at each stage players act synchronously. Assuming that dependencies between actions are fixed (*i.e.*, do not depend on the context), a play is then a Mazurkiewicz trace.

We next define the notion of distributed strategy for a team in such a game. Roughly speaking, a strategy is distributed if any move it predicts for a player only depends on the causal view of that player. In this context, there exist games in which neither team 0 nor team 1 have a winning distributed strategy.

We first show that, as in the sequential framework, one can transform a game G for which team 0 has a distributed strategy with memory μ into a game G^μ for which team 0 has a memoryless strategy. If G and the memory are finite, then so is G^μ . We further transform a distributed game G into a classical 2-players game \tilde{G} , such that team 0 has a memoryless distributed strategy in G if and only if player 0 has a memoryless winning strategy in \tilde{G} . This result is effective: \tilde{G} can be effectively constructed and from a winning strategy of \tilde{G} we can effectively construct a winning distributed strategy for G and vice versa. We then show that if the winning condition is a recognizable trace language, then one can decide whether team 0 has a memoryless distributed winning strategy, and compute it. The restriction to recognizable specifications is not artificial and makes it possible to express a relationship between the architecture of the game and the specification. Moreover, in practice, recognizable languages cover most interesting properties. As in [6,7], specifications depending on the order of independent actions lead to undecidability.

Finally, we explain how to simulate distributed games of [7] in our context. The goals of the two kinds of games are quite different. The aim of [7] is to unify different approaches and to find generic transformations on distributed games (*e.g.*, the reduction of the number of players) to get decidability results, while we first focused on a natural model which we then reduced to sequential games.

2 Preliminaries

In this section, we briefly recall definitions of pomsets and Mazurkiewicz traces. The reader is referred to [3,2] for details.

If (V, \leq) is a poset and $S \subseteq V$, we let $\downarrow S = \{e \in V \mid \exists s \in S, e \leq s\}$. When $e \in V$ then we simply write $\downarrow e$ for $\downarrow \{e\}$ and we let $\downarrow e = \downarrow e \setminus \{e\}$. The successor relation associated with the partial order $<$ is $\ll = < \setminus <^2$.

A *pomset* over an alphabet Σ is a tuple (V, \leq, ℓ) where (V, \leq) is a poset, and $\ell : V \rightarrow \Sigma$ is a mapping called the *labeling*. Elements of V are called *events* or *vertices*. Two pomsets $t = (V, \leq, \ell)$, and $t' = (V', \leq', \ell')$ are *isomorphic*, written $t \sim t'$ if there exists a bijection $\varphi : V \rightarrow V'$ such that $\ell' \circ \varphi = \ell$ and for all $e, f \in V$ $e \leq f$ iff $\varphi(e) \leq' \varphi(f)$. If $\Sigma = \Sigma_1 \times \Sigma_2$ and $\ell(e) = (\ell_1(e), \ell_2(e))$, we write (ℓ_1, ℓ_2) (or even ℓ_1, ℓ_2) instead of ℓ . If $t = (V, \leq, \ell)$ is a pomset, we denote by $\max(t)$

(resp. by $\min(t)$) the set of maximal (resp. minimal) elements of t . The *alphabet* of t is $\text{alph}(t) = \ell(V)$. We let $\text{alphinf}(t) = \{a \in \text{alph}(t) \mid \ell^{-1}(a) \text{ is infinite}\}$.

A *dependence alphabet* is a pair (Σ, D) where Σ is a finite alphabet and D is a reflexive, symmetric binary relation over Σ , called the *dependence relation*. We let $I = \Sigma \times \Sigma \setminus D$ be the independence relation. A (*Mazurkiewicz*) *trace* over (Σ, D) is an isomorphism class of a pomset (V, \leq, ℓ) such that, for all $e, f \in V$: (1) $\ell(e)D\ell(f) \Rightarrow e \leq f$ or $f \leq e$, (2) $e \leq f \Rightarrow \ell(e)D\ell(f)$ and (3) $\downarrow e$ is finite. Two traces t, t' are *independent* if $(\text{alph}(t) \times \text{alph}(t')) \cap D = \emptyset$. We denote by $\mathbb{R}(\Sigma, D)$ (resp. by $\mathbb{M}(\Sigma, D)$) the set of traces (resp. of finite traces) over (Σ, D) . It is well-known that $\mathbb{M}(\Sigma, D)$ is a monoid. The free monoid (resp. the free semigroup) over Σ is denoted by Σ^* (resp. by Σ^+).

A *prefix* of $t = (V, \leq, \ell)$ is a trace (U, \leq, ℓ) , where $U \subseteq V$ satisfies $\downarrow U = U$. We write $s \leq t$ if s is a prefix of t . A *linearization* of t is a labeled total order (V, \preceq, ℓ) such that $e \leq f$ implies $e \preceq f$. For any $w \in \Sigma^*$, there exists a unique trace $[w]$ of which w is a linearization.

3 Distributed games

A distributed system made up of asynchronous processes interacting together and with the environment may be viewed as a single asynchronous model having controllable actions (the system's ones) and uncontrollable actions (the environment's ones). In the game setting, one views actions as players, which are split in two teams Σ_0 (actions of the system) and Σ_1 (actions of the environment). An execution of the system inside the environment corresponds then to a play, a property of the executions to a winning condition, and a distributed controller to a winning distributed strategy for team 0.

If X and I are sets and $J \subseteq I$, then for $x = (x_i)_{i \in I} \in X^I$ we let $x_J = (x_i)_{i \in J} \in X^J$. Given sets $(X_i)_{i \in I}$, and $J \subseteq I$, we let $X_J = \prod_{i \in J} X_i$.

An *architecture* is a tuple $(\Sigma, \mathcal{P}, R, W)$ such that Σ is a finite set of *actions* or *players*, \mathcal{P} is a finite set of *processes*, $R : \Sigma \rightarrow 2^{\mathcal{P}}$ assigns to each $a \in \Sigma$ its read domain $R(a)$, $W : \Sigma \rightarrow 2^{\mathcal{P}}$ assigns to each $a \in \Sigma$ its write domain $W(a)$. We only consider architectures satisfying the following natural restriction, already considered in [14], and sufficient to get a dependence alphabet on actions.

$$\begin{aligned} \forall a \in \Sigma, \quad & \emptyset \neq W(a) \subseteq R(a) \\ \forall a, b \in \Sigma, \quad & R(a) \cap W(b) = \emptyset \iff R(b) \cap W(a) = \emptyset \end{aligned}$$

We define the dependence relation over Σ as $D = \{(a, b) \mid R(a) \cap W(b) \neq \emptyset\}$.

Let $(\Sigma, \mathcal{P}, R, W)$ be an architecture. A *distributed game* over $(\Sigma, \mathcal{P}, R, W)$ is given by a tuple $G = (\Sigma_0, \Sigma_1, (Q_i)_{i \in \mathcal{P}}, (T_a)_{a \in \Sigma}, q^0, \mathcal{W})$ where Σ_0 and Σ_1 are the players of teams 0 and 1 respectively and we have $\Sigma = \Sigma_0 \uplus \Sigma_1$, $\forall i \in \mathcal{P}$, Q_i is the set of local states for process i , $\forall a \in \Sigma$, $T_a \subseteq Q_{R(a)} \times Q_{W(a)}$ gives the local moves of player a , $q^0 \in Q = \prod_{i \in \mathcal{P}} Q_i$ is the *starting position* of G , and \mathcal{W} defines the winning condition of G .

The easiest way to define the semantics of the distributed game is via its sequential game graph whose set of positions is Q , the initial position is q^0 and

there is an a -move from $p \in Q$ to $q \in Q$ (denoted $p \xrightarrow{a} q$) if $(p_{R(a)}, q_{W(a)}) \in T_a$ and $q_{\mathcal{P} \setminus W(a)} = p_{\mathcal{P} \setminus W(a)}$. A sequential play is a sequence $q^0 \xrightarrow{a_1} q^1 \xrightarrow{a_2} q^2 \dots$. Note that in a position $p \in Q$, several players of team 0 and of team 1 may be simultaneously enabled, hence this sequential game graph does not correspond to a conventional (sequential) game in which each position is either a position of player (team) 0 or a position of player (team) 1.

We consider a new symbol $\perp \notin \Sigma$ with $R(\perp) = W(\perp) = \mathcal{P}$ and the alphabet $\Sigma' = \{(a, p) \mid a \in \Sigma \text{ and } p \in Q_{W(a)}\} \cup \{(\perp, q^0)\}$ with the dependence relation $D' = \{((a, p), (b, q)) \mid R(a) \cap W(b) \neq \emptyset\}$. The winning condition of the game is a set of words $\mathcal{W} \subseteq \Sigma'^{\infty}$ which is closed under the usual trace equivalence (see [2,3]). With the sequential play $\pi = q^0 \xrightarrow{a_1} q^1 \xrightarrow{a_2} q^2 \dots$ of G we associate the word $w = (\perp, q^0)(a_1, q_{W(a_1)}^1)(a_2, q_{W(a_2)}^2) \dots$ over Σ' . Note that the word w faithfully encodes the sequential play π and team 0 wins the play π if $w \in \mathcal{W}$.

A better semantics of these distributed games is to view a play directly as a *rooted* trace over the alphabet Σ' . A finite or infinite trace $t = (V, \leq, (\ell, \sigma)) \in \mathbb{R}(\Sigma', D')$ is *rooted* if $\ell^{-1}(\perp) = \{x_\perp\}$ is a singleton and $x_\perp \leq y$ for all $y \in V$. If $s = (U, \leq, (\ell, \sigma))$ is a nonempty prefix of t then $\bar{\sigma}(s) = (\bar{\sigma}(s)_i)_{i \in \mathcal{P}} \in Q$ is defined by $\bar{\sigma}(s)_i = \sigma(y)_i$ where y is the maximal vertex in $\{x \in U \mid i \in W(\ell(x))\}$.

A distributed play of G is a finite or infinite rooted trace $t = (V, \leq, (\ell, \sigma)) \in \mathbb{R}(\Sigma', D')$ such that for each $a \in \Sigma$ and $x \in \ell^{-1}(a)$, we have $(\bar{\sigma}(\downarrow x)_{R(a)}, \sigma(x)) \in T_a$. The winning condition \mathcal{W} is now a subset of $\mathbb{R}(\Sigma', D')$ and team 0 wins the distributed play t if $t \in \mathcal{W}$. The two definitions above are indeed equivalent but the second one is better suited to distributed games and allows a natural definition of a distributed strategy. In sequential games, one often considers infinite plays only. We also consider finite plays because it can be more convenient.

Let $t = (V, \leq, \ell, \sigma) \in \mathbb{R}(\Sigma, D)$ be a rooted trace and let $J \subseteq \mathcal{P}$. The trace $\partial_J t$ is the prefix of t defined by the set of vertices $U = \downarrow\{x \in V \mid W(\ell(x)) \cap J \neq \emptyset\}$.

An asynchronous mapping [1] is a function $\mu : \mathbb{M}(\Sigma, D) \rightarrow M$ such that $\mu(\partial_{A \cup B} t)$ only depends on $\mu(\partial_A t)$ and $\mu(\partial_B t)$, and $\mu(\partial_{R(a)} t.a)$ only depends on $\mu(\partial_{R(a)} t)$ and a . Asynchronous mappings can be computed by deterministic asynchronous transition systems. A *distributed memory* on a game G is an asynchronous mapping $\mu : \mathbb{M}(\Sigma', D') \rightarrow M$. It will be used by players of team 0 as an abstraction (computed in M) of their causal view of the play. A *distributed strategy with memory* μ for team 0 (μ -DS) is a pair (f, μ) where f is a partial function $f : \bigcup_{a \in \Sigma_0} Q_{R(a)} \times M^{R(a)} \times \{a\} \rightarrow Q_{W(a)}$ such that if $f(p, m, a) = q$, then $(p, q) \in T_a$. Intuitively, if $f(p, m, a) = q$, then the strategy f dictates an a -move to $q \in Q_{W(a)}$ when the memory of the play that a can observe using μ is $m \in M^{R(a)}$. If $f(p, m, a)$ is undefined, the a -move is disabled by the strategy. Note that several players of team 0 may be simultaneously enabled by f during a play. In the sequel we write f instead of (f, μ) , μ being understood.

Let f be a μ -DS. Let $\bar{\mu}(t) = (\mu(\partial_i(t)))_{i \in \mathcal{P}}$. A distributed play $t = (V, \leq, \ell, \sigma) \in \mathbb{R}(\Sigma', D')$ is an *f -play* if

$$\forall x \in V, \sigma(x) = f(\bar{\sigma}(\downarrow x)_{R(a)}, \bar{\mu}(\downarrow x)_{R(a)}, a)$$

The play t is f -maximal if $f(\bar{\sigma}(\partial_{R(a)}t)_{R(a)}, \bar{\mu}(\partial_{R(a)}t)_{R(a)}, a)$ is undefined for all $a \in \Sigma_0$ such that $\partial_{R(a)}t$ is finite. The *maximality* condition is natural: if the DS of team 0 dictates some a -moves at some f -play t , then the f -play t is not over and we do not have to decide whether it is winning or not for team 0. Note that this applies also if t is infinite and corresponds to some fairness condition: along an infinite f -play, a move of team 0 cannot be ultimately enabled by f . Observe that any f -play t is the prefix of some f -maximal f -play. If each f -maximal f -play is in \mathcal{W} then f is a *winning* distributed strategy (WDS) for team 0.

A distributed game is not necessarily determined in the sense that it is possible that neither team 0 nor team 1 have a WDS, even with perfect memory. For instance, consider $G = (\Sigma_0, \Sigma_1, (Q_i)_{i \in \mathcal{P}}, (T_a)_{a \in \Sigma}, q^0, \mathcal{W})$ with $\Sigma_0 = \{a\}$, $\Sigma_1 = \{b\}$, $\mathcal{P} = \{1, 2\}$, $R(a) = W(a) = \{1\}$, $R(b) = W(b) = \{2\}$, $Q_1 = Q_2 = \{1\}$, $T_a = Q_1^2$, $T_b = Q_2^2$, $q^0 = (1, 1)$, and $\mathcal{W} = \mathbb{M}(\Sigma', D') \cup \{(\perp, q^0)(a, 1)^\omega(b, 1)^\omega\}$. Assume that team 0 has a DS. If $f((\perp, q^0)(a, 1)^n, a) \neq \emptyset$ for all $n \geq 0$, then team 0 loses if team 1 does not play at all, yielding the play $(\perp, q^0)(a, 1)^\omega$. Conversely, if there exists $n \geq 0$ such that $f((\perp, q^0)(a, 1)^n, a) = \emptyset$, then team 0 loses if team 1 makes infinitely many moves. Symmetrically, team 1 does not have a WDS.

Actually, this non-determinacy is not a problem. For the distributed control problem, we are looking for a WDS allowing controllable events (team 0) to enforce good behaviors but we are not interested in a winning *distributed* strategy for the uncontrollable events: uncontrollable events are played by an environment, and there is no reason to consider only distributed environments.

A *memoryless distributed strategy* (MDS) is a μ -DS with $|\mu(\mathbb{M}(\Sigma', D'))| = 1$, that is, the memory does not record any information. In this case, we write $f(p, a)$ instead of $f(p, m, a)$. A *perfect-memory distributed strategy* is a μ -DS with $\mu(t) = t$. It provides for a move to x with $\ell(x) = a$ the full causal view $\bar{\mu}(\Downarrow x)_{R(a)} = (\partial_i \Downarrow x)_{i \in R(a)}$. Since $\bar{\sigma}(\Downarrow x)_{R(a)}$ can be computed from $(\partial_i \Downarrow x)_{i \in R(a)}$, one can drop the state component in f and write $f(m, a)$ instead of $f(p, m, a)$. As in the sequential case, one can embed a given memory into the game.

Proposition 1. *Let G be a distributed game and let μ be a distributed memory on G . One can construct a distributed game G^μ such that there exists a μ -WDS for G iff there exists a WMDS in G^μ . Moreover, if G is finite and μ is realized by a finite asynchronous automaton, then G^μ is finite.*

Proof. Due to lack of space, we only explain the construction of G^μ and of the strategies. The idea is the same as for sequential games: states of G^μ store states of G as well as distributed memory. Let $G = (\Sigma_0, \Sigma_1, (Q_i)_{i \in \mathcal{P}}, (T_a)_{a \in \Sigma}, q^0, \mathcal{W})$. Then $G^\mu = (\Sigma_0, \Sigma_1, (Q_i \times M)_{i \in \mathcal{P}}, (T_a^\mu)_{a \in \Sigma}, (q^0, m^0), \mathcal{W}^\mu)$. The initial state is given by $m_i^0 = \mu(\varepsilon)$ for $i \in \mathcal{P}$. Let $(a, q) \in \Sigma'$ and $m = (m_i)_{i \in R(a)} \in M^{R(a)}$. If there exists $t = \partial_{R(a)}t \in \mathbb{M}(\Sigma', D')$ such that $\mu(\partial_i t) = m_i$ for all $i \in R(a)$, then we let $m_{R(a)}.(a, q) = \mu(t.(a, q))$ (this is well defined since μ is asynchronous). Otherwise, $m_{R(a)}.(a, q)$ is undefined. We have $((p, m), (p', m')) \in T_a$ iff $(p, p') \in T_a$ and $m' = m.(a, q)$. Finally, a play is winning in G^μ iff the play obtained by removing the memory component is winning in G .

Given a μ -DS f in G , we define a MDS f^μ in G^μ as follows. If $f(p, m, a) = q$, we set $f^\mu((p, m), a) = (q, m.(a, q))$. Conversely, if f^μ is a MDS in G^μ , we define

$f(p, m, a)$ as the first component of $f^\mu((p, m), a)$. One can check that f is winning if and only if f^μ is winning. \square

Lemma 2. *Any f -play t is the prefix of some f -maximal f -play.*

Proof. We proceed by induction on $\text{alphinf}(t)$. If $\text{alphinf}(t) = \Sigma$ then $\partial_{R(a)}t$ is infinite for all $a \in \Sigma_0$ and t is f -maximal. Assume now that $\text{alphinf}(t) \subsetneq \Sigma$. We define a sequence $(t_n)_{n \geq 0}$ by $t_0 = t$ and $t_{n+1} = t_n$ if t_n is f -maximal or $t_{n+1} = t_n \cdot (a, q)$ for some $a \in \Sigma_0$ with $\partial_{R(a)}t_n$ finite and $q \in f(\bar{\sigma}(t_n)_{R(a)}, \bar{\mu}(t_n)_{R(a)}, a)$ otherwise. Then, the increasing sequence $(t_n)_{n \geq 0}$ has a least upper bound $t' = \sqcup_{n \geq 0} t_n$ which is an f -play. Either for some $n \geq 0$ we had t_n f -maximal and $t' = t_n$. Or $\text{alphinf}(t) \subsetneq \text{alphinf}(t')$ and we conclude by induction. \square

4 Global game

In order to use known results of game theory, we want to define a classical two-players global game $\tilde{G} = (Z, T)$ such that team 0 has a WMDS in the distributed game G iff player 0 has a winning memoryless strategy in the global game \tilde{G} .

The positions of the global game are $Z = Z_0 \cup Z_1$ where $Z_0 = Q \times \Sigma_0$ are the positions of player 0 and $Z_1 = Q \times (\Sigma_1 \cup \{0, 1, 2\})$ are the positions of player 1. The initial position is $(q^0, 0) \in Z_1$. In a position (q, a) , the first component describes the current global state of the play and the second component is used both to determine whose turn it is and which action should be executed. The set $T \subseteq (Z_0 \times Z_1) \cup (Z_1 \times Z)$ of moves is defined as follows:

- $(p, b) \rightarrow (p, a)$ with $b \in \{0, 1, 2\}$ and $a \in \Sigma$. Player 1 decides that the next move should be an a -move. In this global game, player 1 is in charge of deciding which actions are used and in which order. This allows him to investigate all possible linearizations of distributed plays.
- $(p, a) \rightarrow (q, 1)$ with $a \in \Sigma$, $(p_{R(a)}, q_{W(a)}) \in T_a$ and $q_{P \setminus W(a)} = p_{P \setminus W(a)}$. This a -move is executed by player 0 or player 1 depending on whether $a \in \Sigma_0$ or $a \in \Sigma_1$.
- $(p, a) \rightarrow (p, 2)$ with $a \in \Sigma_0$. Player 0 *refuses* to make an a -move.
- $(q, b) \rightarrow (q^0, 0)$ with $b \in \{0, 1, 2\}$. These *reset-moves* are used by player 1 to show that player 0 is not following a *distributed* strategy.

Note that player 1 may perform several consecutive moves.

A global play is a finite or infinite sequence $z = z_0 z_1 z_2 \dots \in Z^\infty$ starting from the initial position $z_0 = (q^0, 0)$ and such that $z_n \rightarrow z_{n+1}$ is a move for all $n \geq 0$. Let $z = z_0 z_1 z_2 \dots \in Z^\infty$ be a global play and let $z_n = (q^n, a_n) \in Z$ for $n \geq 0$. We define by induction the sequence $(t_n)_{n \geq 0} \in \mathbb{M}(\Sigma', D')^{\mathbb{N}}$ associated with z . If $z_n = (q^0, 0)$ then $t_n = (\perp, q^0)$. If $a_{n+1} \in \Sigma \cup \{2\}$ then $t_{n+1} = t_n$. Finally, if $a_n = a \in \Sigma$ and $a_{n+1} = 1$ then $t_{n+1} = t_n \cdot (a, q_{W(a)}^{n+1})$. We prove by induction that t_n is a distributed play and $\bar{\sigma}(t_n) = q^n$ for all $n \geq 0$. The only non trivial case is when $a_n = a \in \Sigma$ and $a_{n+1} = 1$. By induction, t_n is a distributed play and $\bar{\sigma}(t_n) = q^n$. We have $(q_{R(a)}^n, q_{W(a)}^{n+1}) \in T_a$ and $q_{R(a)}^n = \bar{\sigma}(\partial_{R(a)}t_n)_{R(a)}$. Therefore, t_{n+1} is a distributed play and using $q_{P \setminus W(a)}^{n+1} = q_{P \setminus W(a)}^n$ we get $\bar{\sigma}(t_{n+1}) = q^{n+1}$.

The global play z is *consistent* if for all $j, k \geq 0$ with $a_j = a_k = a \in \Sigma_0$ and $q_{R(a)}^j = q_{R(a)}^k$ we have $a_{j+1} = a_{k+1}$ and $q_{W(a)}^{j+1} = q_{W(a)}^{k+1}$. The global play z is *fair* if $\{n \geq 0 \mid a_n = 0\}$ is finite and for all $a \in \Sigma_0$, $\{n \geq 0 \mid a_n = a\}$ is infinite. If z is both consistent and fair then we let $N(z) = \max\{n \mid a_n = 0\}$. The sequence $(t_n)_{n \geq N(z)}$ is increasing and admits a least upper bound $t(z)$ which is a distributed play of G .

The winning condition \widetilde{W} of \widetilde{G} only involves infinite plays $z \in Z^\omega$. If z is not consistent then player 0 loses the game since this reveals that he does not mimic a memoryless *distributed* strategy. If z is not fair then player 1 loses the game. Finally, if z is both consistent and fair then player 0 wins the game iff $t(z) \in \mathcal{W}$.

A (global) *strategy* (S) for player 0 in \widetilde{G} is a mapping $g : Z^* Z_0 \rightarrow Z_1$ such that $g(z(p, a)) = (q, b)$ implies $(p, a) \rightarrow (q, b)$. A global play $z = z_0 z_1 z_2 \dots \in Z^\infty$ is *played according to g* (g -play) if each move of player 0 is done according to g : $z_k \in Z_0$ implies $z_{k+1} = g(z_0 \dots z_k)$. If player 0 wins all infinite g -plays then g is a winning strategy (WS) for player 0. A strategy g is *memoryless* if for all $x, x' \in Z^*$ and $y \in Z_0$, we have $g(xy) = g(x'y)$. We write MS and WMS for *memoryless strategy* and *winning memoryless strategy*.

We can now state the main result of this section.

Theorem 3. *The following conditions are equivalent for a distributed game G :*

1. *There exists a WMDS for team 0 in the distributed game G .*
2. *There exists a WMS for player 0 in the global game \widetilde{G} .*
3. *There exists a WS for player 0 in the global game \widetilde{G} .*

The proof of this theorem is split in Proposition 4 for $(3 \Rightarrow 2)$, Proposition 5 for $(1 \Rightarrow 2)$ and Proposition 7 for $(2 \Rightarrow 1)$. The implication $(2 \Rightarrow 3)$ is clear.

Proposition 4. *If player 0 has a winning strategy in $\widetilde{G} = (Z, T)$, then he has a memoryless winning strategy in this game.*

Proof. We write $v \rightarrow_1 w$ if $(v, w) \in T \cap (Z_1 \times Z)$ is a move of player 1 and we let $\overset{*}{\rightarrow}_1$ be the reflexive and transitive closure of \rightarrow_1 . We define the 1-closure of a graph $(Y, S) \subseteq \widetilde{G}$ by $\text{cl}_1(Y, S) = (Y_1, S_1)$, where $Y_1 = \{v \in Z \mid \exists v' \in Y, v' \overset{*}{\rightarrow}_1 v\}$ and $S_1 = S \cup \{(v, w) \in Y_1 \times Y_1 \mid v \rightarrow_1 w\}$.

Let g be a WS for player 0 in $\widetilde{G} = (Z, T)$. We build inductively an increasing sequence x^n of g -plays in \widetilde{G} , and of graphs $\widetilde{G}_n = (Y_n, S_n) \subseteq \widetilde{G}$. Initially, $x^0 = (q^0, 0)$ and $\widetilde{G}_0 = \text{cl}_1(\{(q^0, 0)\}, \emptyset)$. Assume that x^n and $\widetilde{G}_n = \text{cl}_1(\widetilde{G}_n)$ are built and satisfy the following invariants.

- (i) Any position in Y_n is accessible from $(q^0, 0)$ in \widetilde{G}_n .
- (ii) Each move of player 0 in \widetilde{G}_n occurs in x^n : if $(y, y') \in S_n$ and $y \in Z_0$ then we have $x^n = x' y y' x''$ for some $x', x'' \in Z^*$.
- (iii) x^n is a g -play and a path in \widetilde{G}_n ending in $(q^0, 0)$.

If $(\{z\} \times Z) \cap S_n \neq \emptyset$ for all positions $z \in Y_n \cap Z_0$ of player 0, then the construction stops and we let $\overline{G} = (Y, S) = \tilde{G}_n$ and $x = x^n$. Otherwise, choose some $z \in Y_n \cap Z_0$ such that $(\{z\} \times Z) \cap S_n = \emptyset$ and choose by (i) a path $(q^0, 0)yz$ in \tilde{G}_n . Let $x^{n+1} = x^n y z g(x^n y z)(q^0, 0)$ and $\tilde{G}_{n+1} = \text{cl}_1(Y_n \cup \{g(x^n y z)\}, S_n \cup \{(z, g(x^n y z))\})$.

We verify the invariants. Clearly, (i) holds. The only new edge of player 0 in \tilde{G}_{n+1} is $(z, g(x^n y z))$ which occurs in x^{n+1} so (ii) holds as well. By definition, x^{n+1} is a path in \tilde{G}_{n+1} ending in $(q^0, 0)$. In order to prove that $x^{n+1} = x^n y z g(x^n y z)(q^0, 0)$ is a g -play, we only have to show that moves in y are played according to g . So let $y = y' v w y''$ with $v \in Z_0$, $w \in Z_1$ and assume that $x^n y'$ is a g -play. Then, $x^n y' v g(x^n y' v)$ is a g -play and can be extended to an infinite g -play. Since g is a WS, the g -play $x^n y' v g(x^n y' v)$ must be consistent. By (ii), we can write $x^n = x' v w x''$. We deduce that $g(x^n y' v) = w$ by consistency. Hence, the move (v, w) in y is done according to g . Therefore, x^{n+1} is a g -play: (iii) holds.

Note that for all $y \in Y \cap Z_0$, there exists a unique $g'(y) \in Y$ such that $(y, g'(y)) \in S$. The existence follows from the halting condition of the construction. The unicity uses (ii): each move of player 0 in \overline{G} occurs in x which is consistent since it is a g -play. Hence g' extends to a MS as follows: for $x \in Z^*$ and $y = (p, a) \in Z_0$, we let $g'(xy) = g'(y)$ if $y \in Y$ and $g'(xy) = (p, 2)$ otherwise.

We now prove that g' is a WMS for player 0 in \tilde{G} . Let $z \in Z^\omega$ be a g' -play. Since g' is *memoryless*, z is consistent. Using the definition of g' and since $\text{cl}_1(\overline{G}) = \overline{G}$, we deduce that z is a path in \overline{G} . We claim that xz is a g -play. By contradiction, let (v, w) with $v \in Z_0$ be the first move of xz not played according to g . Since x is a g -play, this move occurs necessarily along z : $z = z' v w z''$ and $xz'v$ is a g -play. Since z is a path in \overline{G} , we have $(v, w) \in S$. By (ii) we can write $x = x' v w x''$. Since $x' v w x'' z'v$ is a g -play, its extension by $g(x' v w x'' z'v)$ must be consistent. Hence, $g(x' v w x'' z'v) = w$: a contradiction. Therefore, xz is a g -play and is winning for player 0 since g is a WS. If xz is not fair then z is not fair and is winning for player 0. If xz is fair then we have $t(xz) = t(z)$ since x ends in $(q^0, 0)$. Therefore, $t(z) = t(xz) \in \mathcal{W}$. \square

Proposition 5. *Let f be a deterministic WMDS for team 0 in G . For $(p, a) \in Z_0$, we define $g((p, a)) = (p, 2)$ if $f(p_{R(a)}, a) = \emptyset$ and $g((p, a)) = (q, 1)$ with $q_{\mathcal{P} \setminus W(a)} = p_{\mathcal{P} \setminus W(a)}$ and $f(p_{R(a)}, a) = \{q_{W(a)}\}$ otherwise. Then, g is a WMS for player 0 in the global game \tilde{G} .*

Proof. Consider a g -play $z = z_0 z_1 z_2 \dots \in Z^\omega$. We first show that z is consistent. Let $j, k \geq 0$ with $a_j = a_k = a \in \Sigma_0$ and $q_{R(a)}^j = q_{R(a)}^k$. We have $f(q_{R(a)}^j, a) = f(q_{R(a)}^k, a)$. Since z is a g -play, we deduce from the definition of g that either $a_{j+1} = a_{k+1} = 2$ or $a_{j+1} = a_{k+1} = 1$ and $\{q_{W(a)}^{j+1}\} = \{q_{W(a)}^{k+1}\}$.

If the play is not fair then player 0 wins. Hence, we assume that z is fair. We will show that $t(z)$ is an f -maximal f -play. Since f is a WMDS, it follows that $t(z) \in \mathcal{W}$ and player 0 wins the global play z which concludes the proof.

Let $x \in t(z)$ with $a = \ell(x) \in \Sigma_0$. Let $j > N(z)$ minimal such that $\downarrow x \leq t_{j+1}$. Necessarily, we have $a_j = a$, $\partial_{R(a)} t_j = \downarrow x$, $a_{j+1} = 1$ and $\sigma(x) = q_{W(a)}^{j+1}$. We deduce that $\bar{\sigma}(\downarrow x)_{R(a)} = \bar{\sigma}(t_j)_{R(a)} = q_{R(a)}^j$. Since z is a g -play we have

$f(q_{R(a)}^j, a) = \{q_{W(a)}^{j+1}\}$. Hence, $\sigma(x) \in f(\bar{\sigma}(\downarrow x)_{R(a)}, a)$ and $t(z)$ is an f -play. It remains to check that it is f -maximal. Let $a \in \Sigma_0$ such that $\partial_{R(a)}t(z)$ is finite. Since z is fair, there exists $k > N(z)$ with $a_k = a$ and $\partial_{R(a)}t_k = \partial_{R(a)}t(z)$. Since $t_{k+1} \leq t(z)$ it follows that $a_{k+1} = 2$ (player 0 refused the a -move). We have $\bar{\sigma}(\partial_{R(a)}t(z))_{R(a)} = \bar{\sigma}(\partial_{R(a)}t_k)_{R(a)} = q_{R(a)}^k$. Since z is a g -play, we deduce that $f(\bar{\sigma}(\partial_{R(a)}t(z))_{R(a)}, a) = f(q_{R(a)}^k, a) = \emptyset$. \square

To prove the implication $(2 \Rightarrow 1)$ of Theorem 3, we exploit reset-moves.

Lemma 6. *Let g be a WMS of player 0 in the global game \tilde{G} . Let $(p^1, a) \in Z_0$ and $(p^2, a) \in Z_0$ be accessible in g -plays and such that $p_{R(a)}^1 = p_{R(a)}^2$. Then, $g(p^1, a) = (p^1, 2)$ iff $g(p^2, a) = (p^2, 2)$ and if $g(p^1, a) = (q^1, 1)$ and $g(p^2, a) = (q^2, 1)$ then $q_{W(a)}^1 = q_{W(a)}^2$.*

Proof. Let $x = x_0x_1x_2 \cdots \in Z^\omega$ be a g -play and let $k > 0$ with $x_k = (p^1, a)$. Let $y = y_0y_1y_2 \cdots \in Z^\omega$ be a g -play and let $n > 0$ with $y_n = (p^2, a)$. Then, $z = x_0 \cdots x_k x_{k+1} y_0 y_1 y_2 \cdots \in Z^\omega$ is also a g -play (note that $x_{k+1} \rightarrow y_0$ is a reset-move). Since g is a WMS, z must be consistent. Hence, $x_{k+1} = (p^1, 2)$ iff $y_{n+1} = (p^2, 2)$ and if $x_{k+1} = (q^1, 1)$ iff $y_{n+1} = (q^2, 1)$ then $q_{W(a)}^1 = q_{W(a)}^2$. Since z is a g -play, we have $x_{k+1} = g(x_k)$ and $y_{n+1} = g(y_n)$. \square

Proposition 7. *Let g be a WMS of player 0 in the global game \tilde{G} . For $(p, a) \in Z_0$ accessible by a g -play, we define $f(p_{R(a)}, a) = \emptyset$ if $g(p, a) = (p, 2)$ and $f(p, a) = \{q_{W(a)}\}$ if $g(p, a) = (q, 1)$. Then, f is a WMDS of team 0 in the distributed game G .*

Proof. Note first that f is well-defined by Lemma 6.

Let t be an f -maximal f -play and let $(\perp, q^0)(a_1, p^1)(a_2, p^2) \cdots$ be a linearization of t . For $n \geq 0$, let $t[n] = (\perp, q^0)(a_1, p^1) \cdots (a_n, p^n)$ and let $q^n = \bar{\sigma}(t[n])$. Define the sequence $(z_n)_{n \geq 0}$ by $z_0 = (q^0, 0)$ and for $n \geq 0$, $z_{2n+1} = (q^n, a_{n+1})$ and $z_{2n+2} = (q^{n+1}, 1)$. We show that $z = z_0 z_1 z_2 \cdots \in Z^\omega$ is a g -play. Let $n \geq 0$. We have to show that $z_{2n+1} \rightarrow z_{2n+2}$ is a g -move. We have $q_{R(a)}^n = \bar{\sigma}(\partial_{R(a)}t[n])_{R(a)}$ and since t is an f -play we get $\{p^{n+1}\} = f(\bar{\sigma}(\partial_{R(a)}t[n])_{R(a)}, a) = f(q_{R(a)}^n, a)$. By definition of f , we have $g(q^n, a) = (q, 1)$ with $q_{W(a)} = p^{n+1}$ and $q_{\mathcal{P} \setminus W(a)} = q_{\mathcal{P} \setminus W(a)}^n$. Since we also have $q_{W(a)}^{n+1} = \bar{\sigma}(t[n+1])_{W(a)} = p^{n+1}$ and $q_{\mathcal{P} \setminus W(a)}^{n+1} = \bar{\sigma}(t[n+1])_{\mathcal{P} \setminus W(a)} = \bar{\sigma}(t[n])_{\mathcal{P} \setminus W(a)} = q_{\mathcal{P} \setminus W(a)}^n$, we deduce that $q^{n+1} = q$ and $(q^n, a) \rightarrow (q^{n+1}, 1)$ is a g -move.

Note that z is not necessarily fair so we will transform it into a fair g -play. Let $A = \text{alphinf}(t)$ and $B = \{b_1, \dots, b_k\}$ be the set of letters in Σ_0 such that $\partial_{R(a)}t$ is finite. Let $N > 0$ such that $\text{alph}(t[N]^{-1}t) = \text{alphinf}(t)$. We have $A \times B \subseteq I$ hence $\partial_{R(b)}t = \partial_{R(b)}t[n]$ for all $n \geq N$ and $b \in B$. For $n \geq N$, let $y_n = (q^n, b_1)(q^n, 2) \cdots (q^n, b_k)(q^n, 2)$. For all $b \in B$, since t is f -maximal we have $f(q_{R(b)}^n, b) = f(\bar{\sigma}(\partial_{R(b)}t)_{R(b)}, b) = \emptyset$. We deduce using the definition of f that $g(q^n, b) = (q^n, 2)$ for all $b \in B$ which shows that each move of player 0 in y_n is done according to g . Therefore, $x = z_0 z_1 \cdots z_{N-1} z_N y_N z_{N+1} y_{N+1} \cdots$ is a g -play.

We have $\Sigma_0 \subseteq A \cup B$ which implies that x is fair. Since g is a WMS for player 0, we deduce that $t = t(z) = t(x) \in \mathcal{W}$. \square

Even if \mathcal{W} is rational ($\mathcal{W} = [L]$, where $L \in \text{Rat}(\Sigma^*)$), determining if team 0 has a W(M)DS is undecidable. Indeed, on $\mathbb{M}(\Sigma, D) = A^* \times B^*$, determining if a rational trace language \mathcal{L} is $[\Sigma^*]$ is undecidable [2]. From such a language \mathcal{L} , we construct a 2-processes game in which team 0 has a WMDS iff $\mathcal{L} = [\Sigma^*]$: $\Sigma_0 = \emptyset$, $\Sigma_1 = A \uplus B$, $R(a) = W(a) = \{1\}$ for $a \in A$ and $R(b) = W(b) = \{2\}$ for $b \in B$. Finally, $|Q| = 1$ (so that we identify Σ' and Σ), and $\mathcal{W} = \mathcal{L} \cup (\mathbb{R}(\Sigma, D) \setminus \mathbb{M}(\Sigma, D))$. Players of team 1 nondeterministically choose a move in some finite local game, so that any possible trace is a play. Now, team 0 has a WMDS iff he has a WDS iff team 1 cannot generate a finite trace outside \mathcal{L} , that is, iff $\mathcal{L} = [\Sigma^*]$.

We now explain how to use Theorem 3 to decide if team 0 has a WDS. Denote by $\text{Lin}(t)$ the set of all linearizations of $t \in \mathbb{R}(\Sigma, D)$. Properties considered in practice are recognizable (*i.e.*, $\text{Lin}(\mathcal{W})$ is rational), and as noted in [6], there are many temporal logics expressing only recognizable specifications. To determine whether team 0 has a WMDS in G with \mathcal{W} recognizable, we could enumerate all memoryless distributed strategies for player 0, and check whether one is winning. This amounts to testing an inclusion between recognizable trace languages. Theorem 3 provides a better algorithm. The principle is to build the global game \tilde{G} , to transform it into a parity game and to apply known algorithms.

Let \mathcal{A} be a parity automaton accepting $\text{Lin}(\mathcal{W})$. The winning condition $\tilde{\mathcal{W}}$ for player 0 on the global game \tilde{G} can be defined by $\tilde{\mathcal{W}} = \tilde{\mathcal{W}}_c \cap (\text{Lin}(\mathcal{W}) \cup \tilde{\mathcal{W}}_{\text{nf}})$, where $\tilde{\mathcal{W}}_c = \{z \in Z^\omega \mid z \text{ is consistent}\}$, and $\tilde{\mathcal{W}}_{\text{nf}} = \{z \in Z^\omega \mid z \text{ is not fair}\}$. We describe informally how to construct a parity automaton accepting $\tilde{\mathcal{W}}$. One can build a Büchi automaton accepting consistent plays in Z^ω : it records all transitions $(p_{R(a)}, q_{W(a)})$ performed by player 0, as well as the refused transitions. It falls into the unique rejecting state as soon as an inconsistent move is detected. One can also build a parity automaton checking that a play, supposed consistent, is not fair, by checking that there is an infinite number of reset moves (a Büchi condition) or that, for some $a \in \Sigma_0$, there is only a finite number of states of the form (p, a) (co-Büchi conditions). From the parity automaton \mathcal{A} and from these automata, it should now be clear how to build a parity automaton for $\tilde{\mathcal{W}}$.

Observe that using Proposition 1, one can also determine whether team 0 has a μ -WDS for a given finite distributed memory μ .

5 Related approaches

A distributed game $G = \langle P, E, \text{Tr}, \text{Acc}, q^0 \rangle$ as in [7] is built from n local games G_1, \dots, G_n , where $G_i = \langle P_i, E_i, \text{Tr}_i, q_i^0 \rangle$ with $\text{Tr}_i \subseteq (P_i \times E_i)$. Positions of the environment are $E = \prod_i E_i$. The position set of the players is $P = \prod_i (P_i \cup E_i) \setminus E$. Transitions of the players are defined with a cartesian product: $\text{Tr}_p = (\prod_i (\text{Tr}_i \cup \Delta_i)) \cap (P \times E)$ where $\Delta_i = \{(x_i, x_i) \mid x_i \in E_i\}$ is the diagonal. Transitions of the environment are simply given by a subset Tr_e of $E \times P$, and $\text{Tr} = \text{Tr}_e \uplus \text{Tr}_p$. A play of G starts in position $q^0 \in E$, and moves from the environment and from

the players alternate. Hence, any infinite play is in $(E \cdot P)^\omega$, and the winning condition is a subset $\text{Acc} \subseteq (E \cdot P)^\omega$.

There is a natural translation from these games to our setting. With G , we associate the distributed game $\bar{G} = (\Sigma_0, \Sigma_1, (T_a)_{a \in \Sigma}, q^0, \mathcal{W})$ as follows. The set of processes is $\mathcal{P} = \{1, \dots, n\}$ and the local states are $Q_i = P_i \cup E_i$ for $i \in \mathcal{P}$. Team 0 is defined by $\Sigma_0 = \{1, \dots, n\}$ with $R(i) = W(i) = \{i\}$ for all $i \in \Sigma_0$. The transitions for player i are simply $T_i = \text{Tr}_i$. Team 1 consists of a single player e (the environment) with $R(e) = W(e) = \mathcal{P}$. Its transitions are $T_e = \text{Tr} \cap (E \times P)$.

To define the winning condition \mathcal{W} , we associate with each infinite play $w = e^0 x^1 e^1 \dots \in (E \cdot P)^\omega$ of G with $e^0 = q^0$ a distributed play trace $(w) \in \mathbb{R}(\Sigma', D')$ as the least upper bound of $(t_i)_{i \geq 0}$ where the increasing sequence $(t_n)_{n \geq 0}$ is defined inductively by $t_0 = (\perp, q^0)$ and $t_{n+1} = t_n \cdot (e, x^{n+1}) \cdot \prod_{i | x_i^{n+1} \in P_i} (i, e_i^{n+1})$. Finally, the winning condition is $\mathcal{W} = \text{trace}(\text{Acc}) \cup \{t \in \mathbb{M}(\Sigma', D') \mid \bar{\sigma}(t) \in E\}$.

The distributed games of [7] are thus a special case of our games in which all players of team 0 are completely local and the environment consists of a single global player. Note that, in the game G , information between players can only flow through environment moves and the environment can decide which information is exchanged between players.

However, the crucial difference between games of [7] and the ones presented here concerns the definition of strategies. In [7], a strategy is a tuple of mappings $f_i : (E_i P_i)^+ \rightarrow E_i$. Hence a move of player i only depends on its *local* view consisting of the history on its process only. In our setting, the strategy of player i which is the mapping $f(-, i)$ is based on its causal view $\partial_{R(i)} t$. Since actions of the environment are global, the causal view is almost the complete global view of the game. It is therefore clear that if there exists a WS for the players in G , then there exists a WDS for team 0 in \bar{G} . The converse is false: it is easy to find a game G not determined while there is a winning strategy for team 0 in \bar{G} . Yet, one can prove that if a game G of [7] is determined, then there is a WDS for team 0 in \bar{G} iff players have a WS in G .

If we want to get an equivalence, we have to restrict the memory used by our strategies to the local view, that is, to change the notion of memory used by the strategies. The i -projection of $t \in \mathbb{M}(\Sigma, D')$ is $\Pi_i(t)$ where Π_i is the morphism from $\mathbb{M}(\Sigma', D')$ to Q_i^* defined by $\Pi_i(x, q) = q_i$ if $x = e$ or $x = i$ and $\Pi_i(x, q) = \varepsilon$ otherwise. Since player i is only aware of move he takes part in, we have to abstract away from unobservable stuttering of the environment. For this we use the congruence on Q_i^* generated by $p_i^? = p_i$ for $p_i \in E_i$ and we write w_{\equiv} for the equivalence class of $w \in Q_i^*$. We say that a distributed strategy f is local if for all $i \in \Sigma_0$, $f(t, i)$ depends only on i and $\Pi_i(t)_{\equiv}$.

Proposition 8. *The players have a WS in G iff team 0 has a local WDS in \bar{G} .*

In the distributed control problem presented in [6], the environment performs local moves and the transitions of a controllable action is defined by a cartesian product of local transition functions. It is therefore straightforward to translate as above these games to our framework. Since local strategies are used in [6], Propositions 8 also holds.

6 Alternating asynchronous automata (AAA)

Another application of distributed games is to define the semantics and study alternating asynchronous automata.

Given a finite set Q , $\mathcal{B}^+(Q)$ is the set of boolean positive formulae over Q , that is, formulae built from elements of Q using the logical connectives \vee and \wedge . We disallow the empty conjunction and the empty disjunction in formulae. A subset X of Q *satisfies* $\varphi \in \mathcal{B}^+(Q)$, written $X \models \varphi$, if the assignment χ defined by $\chi(q) = \mathbf{true}$ iff $q \in X$ satisfies φ . Note that since the empty conjunction is not allowed in formulae, $X \models \varphi$ implies that X is nonempty.

An *alternating asynchronous automaton* over an architecture $(\Sigma, \mathcal{P}, R, W)$ is a tuple $\mathcal{A} = ((Q_i)_{i \in \mathcal{P}}, (\delta_a)_{a \in \Sigma}, q^0, \mathcal{F})$ where $\forall i \in \mathcal{P}$, Q_i is the *set of local states* for process i , $\forall a \in \Sigma$, $\delta_a : Q_{R(a)} \rightarrow \mathcal{B}^+(Q_{W(a)})$ is the *local transition function* of action a , $q^0 \in Q = \prod_{i \in \mathcal{P}} Q_i$ is the *initial state*, and $\mathcal{F} \subseteq Q$ is the accepting condition.

Pomsets with a binary conflict relation, introduced in [8] are called *prime event structures*. In this paper, we simply call them *event structures*.

An *event structure* over Σ is a tuple $\rho = (E, \leq, \#, \ell)$, where (E, \leq) is a pomset and $\#$ is a binary symmetric and irreflexive relation on the event set E , called the *conflict relation*, satisfying the following inheritance property:

$$(e \# f \text{ and } f \leq g) \implies e \# g \quad (1)$$

An *initial conflict* is a pair (e, f) such that $e \# f$ and for all $e' \leq e$, $f' \leq f$, $e' \# f' \implies e' = e, f' = f$. We write $e \#_i f$ if (e, f) is an initial conflict. A *configuration* of ρ is a subset C of E such that $\downarrow C = C$ and $C \times C \cap \# = \emptyset$. We denote by $\mathcal{C}_{\max}(\rho)$ the set of maximal configurations of ρ .

A *run* of the AAA \mathcal{A} on a finite trace $t \in \mathbb{M}(\Sigma, D)$ is an event structure $\rho = (E, \leq, \#, \ell, \sigma)$ with $\ell : E \rightarrow \Sigma \cup \{\perp\}$ and $\sigma : E \rightarrow \bigcup_{J \subseteq \mathcal{P}} Q_J$ satisfying the following conditions. First, $\ell^{-1}(\perp) = \{e_\perp\}$ is a singleton and we have

$$\forall e \in E, \quad e_\perp \leq e \quad (2)$$

$$\forall e, e' \in E, \quad e \#_i e' \iff e \neq e' \ \& \ \ell(e) = \ell(e') \ \& \ \downarrow e = \downarrow e' \quad (3)$$

$$\forall C \in \mathcal{C}_{\max}(\rho), \quad (C, \leq, \ell) \sim \perp \cdot t \quad (4)$$

The set of siblings of $e \in E$ is $H(e) = \{e' \in E \mid \ell(e') = \ell(e) \ \& \ \downarrow e' = \downarrow e\}$. Second, $\sigma(e) \in Q_{W(\ell(e))}$ for all $e \in E$. Hence, for a nonempty configuration $C \subseteq E$, we can define $\bar{\sigma}(C) \in Q$ by $\bar{\sigma}(C)_i = \sigma(e)_i$ where e is the (unique) maximal event in C such that $i \in W(\ell(e))$. The last condition is

$$\forall e \in E, \quad \sigma(H(e)) \models \delta_{\ell(e)}(\bar{\sigma}(\downarrow e)_{R(\ell(e))}) \quad (5)$$

The run ρ is accepting if $\bar{\sigma}(C) \in \mathcal{F}$ for all $C \in \mathcal{C}_{\max}(\rho)$. The AAA \mathcal{A} accepts a trace $t \in \mathbb{M}(\Sigma, D)$ if there exists an accepting run of \mathcal{A} on t . We denote by $\mathcal{L}(\mathcal{A})$ the set of traces accepted by \mathcal{A} .

A run $\rho = (E, \leq, \#, \ell, \sigma)$ is *reduced* if for all $e \in E \setminus \{e_\perp\}$ and $e' \in H(e)$, if $\sigma(e) = \sigma(e')$ then $e = e'$. Note that $t \in \mathcal{L}(\mathcal{A})$ iff there exists a reduced accepting run of \mathcal{A} on t . Reduced runs are interesting due to the following lemma that will be used later.

Lemma 9. *Let C, C' be two configurations of a reduced run ρ and let $\varphi : (C, \leq, \ell, \sigma) \rightarrow (C', \leq, \ell, \sigma)$ be an isomorphism. Then, φ is the identity and $C = C'$.*

Proof. Let $\varphi : C \rightarrow C'$ be an isomorphism. We use an induction on $|C|$. If $|C| = 1$ then $C = \{e_\perp\} = C'$ and φ must be the identity. Now, assume that $|C| > 1$ and let $e \in C$ be maximal. Let $e' = \varphi(e)$. Since φ is an isomorphism, e' is maximal in C' and both $C \setminus \{e\}$ and $C' \setminus \{e'\}$ are configurations of ρ . Moreover, φ is an isomorphism from $C \setminus \{e\}$ to $C' \setminus \{e'\}$. By induction, we get $C \setminus \{e\} = C' \setminus \{e'\}$ and φ is the identity from $C \setminus \{e\}$ to $C' \setminus \{e'\}$. We deduce that $\Downarrow e = \Downarrow e'$. Using again that φ is an isomorphism we have $\ell(e) = \ell(e')$ and $\sigma(e) = \sigma(e')$. Therefore $e' \in H(e)$ and since ρ is reduced we get $e' = e$. \square

An alternative definition of the semantics of AAA's is to use distributed games. Let $\Gamma_0 = \{\dot{a} \mid a \in \Sigma\}$ and $\Gamma_1 = \{\ddot{a} \mid a \in \Sigma\}$ be two copies of Σ and let $\Gamma = \Gamma_0 \uplus \Gamma_1$. For $a \in \Sigma$, we define $R(\dot{a}) = R(\ddot{a}) = R(a)$ and $W(\dot{a}) = W(\ddot{a}) = W(a)$. Note that the architecture $(\Gamma, \mathcal{P}, R, W)$ also satisfies condition (3). We associate with an AAA $\mathcal{A} = ((Q_i)_{i \in \mathcal{P}}, (\delta_a)_{a \in \Sigma}, q^0, \mathcal{F})$ over the architecture $(\Sigma, \mathcal{P}, R, W)$ a distributed game $G = (\Gamma_0, \Gamma_1, (S_i)_{i \in \mathcal{P}}, (T_a)_{a \in \Gamma}, s^0, \mathcal{W})$ over the architecture $(\Gamma, \mathcal{P}, R, W)$ as follows:

- $S_i = Q_i \cup \bigcup_{a \mid i \in W(a)} (2^{Q_{W(a)}} \times \{\ddot{a}\})$.
- $s^0 = q^0$.
- A move of player \dot{a} from team 0 is only possible from a state of the form $q \in Q_{R(a)}$. Then, player \dot{a} chooses a (nonempty) set $X \subseteq Q_{W(a)}$ such that $X \models \delta_a(q)$ and writes the pair (X, \ddot{a}) on every process $i \in W(a)$:

$$T_{\dot{a}} = \{(q, \xi) \mid q \in Q_{R(a)}, \xi = (\xi_i)_{i \in W(a)} \text{ and} \\ \exists X \subseteq Q_{W(a)}, X \models \delta_a(q) \ \& \ \xi_i = (X, \ddot{a}), \forall i \in W(a)\}.$$

- A move of player \ddot{a} from team 1 is only possible from a tuple $(\xi_i)_{i \in R(a)}$ if there exists a nonempty set $X \subseteq Q_{W(a)}$ such that for all $i \in W(a)$ we have $\xi_i = (X, \ddot{a})$. Then, player \ddot{a} chooses an element $q \in X$ and writes q_i on every process $i \in W(a)$:

$$T_{\ddot{a}} = \{(\xi, q) \mid \xi = (\xi_i)_{i \in R(a)}, q \in Q_{W(a)} \text{ and} \\ \exists X \subseteq Q_{W(a)}, q \in X \ \& \ \xi_i = (X, \ddot{a}), \forall i \in W(a)\}.$$

The winning condition depends on the problem on AAA's we consider. As an example, we consider the acceptance problem: given a trace $t \in \mathbb{M}(\Sigma, D)$ we want to know whether t is accepted by \mathcal{A} . Define $\Gamma'_0 = \{(\dot{a}, \xi) \mid \xi \in S_{W(a)}\}$, $\Gamma'_1 = \{(\ddot{a}, q) \mid q \in S_{W(a)}\}$, $\Gamma' = \Gamma'_0 \cup \Gamma'_1 \cup \{(\perp, q^0)\}$ and let

$$K = \{(\dot{a}, \xi)(\ddot{a}, q) \in \Gamma'_0 \Gamma'_1 \mid \exists X \subseteq Q_{W(a)}, q \in X \ \& \ \xi_i = (X, \ddot{a}), \forall i \in W(a)\}$$

and $\Pi : K^* \rightarrow \mathbb{M}(\Sigma, D)$ be defined by $\Pi((\dot{a}, \xi)(\ddot{a}, q)) = a$. For $s \in K^*$, we define as usual $\bar{\sigma}((\perp, q^0) \cdot s) \in Q$ by $\bar{\sigma}((\perp, q^0) \cdot s)_i = \sigma(e)_i$ if e is the last vertex of $(\perp, q^0) \cdot s$ that has written on process i .

Theorem 10. Let $\mathcal{W}_t = \mathcal{W}_t^1 \cup \mathcal{W}_t^2$ with

$$\mathcal{W}_t^1 = \{(\perp, q^0) \cdot s \mid s \in K^*, \Pi(s) = t \text{ \& } \bar{\sigma}((\perp, q^0) \cdot s) \in \mathcal{F}\},$$

$$\mathcal{W}_t^2 = \{(\perp, q^0) \cdot s \mid \exists s' \in \Gamma_1^{'+}, ss' \in K^* \text{ \& } \Pi(ss') \leq t\}.$$

Then, $t \in \mathcal{L}(\mathcal{A})$ iff team 0 has a WDS in $G_t = (\Gamma_0, \Gamma_1, (S_i)_{i \in \mathcal{P}}, (T_a)_{a \in \Gamma}, s^0, \mathcal{W}_t)$.

Before proving this result, let us point out that the game G_t is not necessarily determined: it can happen that neither team 0 nor team 1 has a WDS. Consider the following alternating (cellular) AAA on 2 processes, with actions a and b where a (resp. b) reads and writes on process 1 (resp. 2). The initial state is $(0, 0)$. Transitions are given by $\delta_a(0) = 1 \wedge 2$, $\delta_a(1) = 3 \vee 5$, $\delta_a(2) = 4 \vee 6$, $\delta_b(0) = 1 \wedge 2$, $\delta_b(1) = 3 \vee 5$, $\delta_b(2) = 4 \vee 6$. The accepting set is $\mathcal{F} = \{(i, j) \mid i = j \text{ or } i - j \equiv 1 \pmod{4}\}$. Then the game G_{aabb} is not determined. First, one checks that the trace $aabb$ is not accepted by the automaton, by considering the 16 different runs on $aabb$. These runs only differ by the labeling under σ of the maximal events, and in all cases, there is a configuration C such that $\bar{\sigma}(C) \notin \mathcal{F}$. By Theorem 10, team 0 does not have a WDS. Consider then a DS for team 1. A play begins with a move $(\dot{a}, \{1, 2\})$ or $(\dot{b}, \{1, 2\})$ of team 0. Team 1 has to reply, say by \ddot{a} , and by symmetry we can assume that it chooses $(\ddot{a}, 1)$. Team 0 next chooses $(\dot{a}, \{3\})$. Next, if team 1 chooses $(\ddot{b}, 1)$, team 0 will reply with $(\dot{b}, \{3\})$ whereas if team 1 chooses $(\ddot{b}, 2)$, it replies with $(\dot{b}, \{6\})$.

In order to prove Theorem 10, our first aim is to show that if team 0 has a WDS in G_t then t is accepted by \mathcal{A} (Lemmas 11, 12 and 13).

Lemma 11. Let f be a WDS for team 0 in the game G_t and let $s = (\perp, q^0)s'$ be an f -play with $s' \in K^*$. Then, $\Pi(s') \leq t$ and $a \in \min(\Pi(s')^{-1}t)$ iff $f(\partial_{R(a)}s, \dot{a}) \neq \emptyset$.

Proof. By Lemma 2, we find an f -maximal f -play s'' with $s \leq s''$. Since f is a WDS we have $s'' \in \mathcal{W}_t$ and we deduce that $\Pi(s') \leq t$.

Let $a \in \min(\Pi(s')^{-1}t)$. Note that $\Pi(s') < t$ hence $s \notin \mathcal{W}_t$. We show that $f(\partial_{R(a)}s, \dot{a}) \neq \emptyset$ by induction on $|\Pi(s')^{-1}t|$. Since f is a WDS and $s \notin \mathcal{W}_t$, the f -play s is not f -maximal. Let $\dot{c} \in \Gamma_0$ such that $\partial_{R(c)}s$ is finite and $f(\partial_{R(c)}s, \dot{c}) \neq \emptyset$. If $c = a$ then we are done. Otherwise, we let $\xi \in f(\partial_{R(c)}s, \dot{c})$. We have $\bar{\sigma}(s)_{R(c)} = \bar{\sigma}(\partial_{R(c)}s)_{R(c)}$ hence $(\bar{\sigma}(s)_{R(c)}, \xi) \in T_{\dot{c}}$ and we deduce that $\xi = (X, \ddot{c})_{i \in W(c)}$ for some $X \models \delta_c(\bar{\sigma}(s)_{R(c)})$. Note that therefore, X is nonempty, and let $q \in X$. We have $(\xi, q) \in T_{\ddot{c}}$. Hence, $r' = s'(\dot{c}, \xi)(\ddot{c}, q) \in K^*$ and $r = (\perp, q^0)r'$ is an f -play. Using the first point, we have $\Pi(r') \leq t$. Since $\Pi(r') = \Pi(s')c$ and $a \neq c$ we deduce that a and c are independent, so $\partial_{R(a)}r = \partial_{R(a)}s$ and $a \in \min(\Pi(r')^{-1}t)$. We have $|\Pi(r')^{-1}t| < |\Pi(s')^{-1}t|$ hence by induction we deduce that $f(\partial_{R(a)}r, \dot{a}) \neq \emptyset$. Therefore, $f(\partial_{R(a)}s, \dot{a}) \neq \emptyset$.

Conversely, assume that $\xi \in f(\partial_{R(a)}s, \dot{a})$ for some $a \in \Sigma$. Then, $s(\dot{a}, \xi)$ is also an f -play. By Lemma 2 it is a prefix of an f -maximal f -play $s(\dot{a}, \xi)s''$ which must be in \mathcal{W}_t since f is a WDS. If $s(\dot{a}, \xi)s'' \in \mathcal{W}_t^1$ then $\Pi(s'(\dot{a}, \xi)s'') = t$. If $s(\dot{a}, \xi)s'' \in \mathcal{W}_t^2$ then we find $s''' \in \Gamma_1^{'+}$ such that $\Pi(s'(\dot{a}, \xi)s''s''') \leq t$. In both cases we obtain $a \in \min(\Pi(s')^{-1}t)$. \square

We introduce some notation. Let $\rho = (E, \leq, \#, \ell, \sigma)$ be a run of \mathcal{A} and $e \in E$ with $a = \ell(e)$. We define $\nu(e) = (\xi_i)_{i \in W(a)}$ with $\xi_i = (\sigma(H(e)), \ddot{a})$ for all $i \in W(a)$. Now, a trace $s = (V, \leq, \ell, \sigma) \in (\perp, q^0)K^*$ is associated with a configuration C of ρ , denoted $C \approx s$, if there is a bijection $\cdot : C \rightarrow V \setminus \ell^{-1}(\Gamma_0)$ such that

- $\forall e_1, e_2 \in C, e_1 \leq e_2$ iff $\ddot{e}_1 \leq \ddot{e}_2$,
- $\forall e \in C \setminus \{e_\perp\}, \ell(e) = a$ implies $\ell(\ddot{e}) = \ddot{a}$; $\sigma(\ddot{e}) = \sigma(e)$; and $\sigma(\dot{e}) = \nu(e)$ where $\dot{e} < \ddot{e}$ is the unique immediate predecessor of \ddot{e} .

Lemma 12. *Let f be a WDS for team 0 in the game G_t . For each $r \leq t$, there exists a run ρ_r of \mathcal{A} on r such that for all $C \in \mathcal{C}_{\max}(\rho_r)$, there exists an f -play $s \in (\perp, q^0)K^*$ such that $C \approx s$.*

Proof. We proceed by induction on r . If $r = \varepsilon$ is the empty trace then the unique run ρ_ε on the empty trace suits our needs. Now, assume that $ra \leq t$ and that we have constructed a run $\rho_r = (E, \leq, \#, \ell, \sigma)$ fulfilling the requirements above.

Let $\chi = \{\partial_{R(a)}C \mid C \in \mathcal{C}_{\max}(\rho_r)\}$. For each $B \in \chi$, we define a set $X_B \models \delta_a(\bar{\sigma}(B)_{R(a)})$ as follows. Let $C \in \mathcal{C}_{\max}(\rho_r)$ such that $B = \partial_{R(a)}C$. Let $s = (\perp, q^0)s'$ with $s' \in K^*$ and $C \approx s$. Using (4) we get $\Pi(s') = r < t$. Hence by Lemma 11 we deduce that $f(\partial_{R(a)}s, \dot{a}) \neq \emptyset$. Let $\xi_B = (X_B, \ddot{a})_{i \in W(a)} \in f(\partial_{R(a)}s, \dot{a})$. Let $p = \bar{\sigma}(s)_{R(a)} = \bar{\sigma}(\partial_{R(a)}s)_{R(a)}$. We have $(p, \xi) \in T_{\dot{a}}$ hence $X_B \models \delta_a(p)$. We have $B = \partial_{R(a)}C \approx \partial_{R(a)}s$ hence we get $\bar{\sigma}(B)_{R(a)} = p$ and $X_B \models \delta_a(\bar{\sigma}(B)_{R(a)})$.

Note that the set X_B is well defined, i.e., it does not depend on the chosen maximal configuration C and trace s . Indeed, let $C_1, C_2 \in \mathcal{C}_{\max}(\rho_r)$ be such that $\partial_{R(a)}C_1 = \partial_{R(a)}C_2$. Let $s_1, s_2 \in (\perp, q^0)K^*$ such that $C_1 \approx s_1$ and $C_2 \approx s_2$. Then, $\partial_{R(a)}s_1 \approx \partial_{R(a)}C_1 = \partial_{R(a)}C_2 \approx \partial_{R(a)}s_2$ and we deduce that $\partial_{R(a)}s_1 = \partial_{R(a)}s_2$.

Consider $E' = E \uplus (\bigsqcup_{B \in \chi} X_B)$. We extend $<, \#, \ell$ and σ to E' as follows. We set $e_1 < e_2$ if $e_1 \in B$ and $e_2 \in X_B$ for some $B \in \chi$. We set $e_1 \# e_2$ if $e_1 \neq e_2$ and $e_1, e_2 \in X_B$ for some $B \in \chi$. Finally, we set $\ell(e) = a$ and $\sigma(e) = e$ if $e \in X_B$ for some $B \in \chi$.

We first show that for $B_1, B_2 \in \chi$ with $B_1 \neq B_2$ we have $X_{B_1} \times X_{B_2} \subseteq \#$. For $i = 1, 2$, let $C_i \in \mathcal{C}_{\max}(\rho_r)$ be such that $B_i = \partial_{R(a)}C_i$. We have $(C_i, \leq, \ell) \sim \perp \cdot r$ hence we deduce that $(B_1, \leq, \ell) \sim \perp \cdot \partial_{R(a)}r \sim (B_2, \leq, \ell)$. Let $e_1 \in B_1 \setminus B_2$ be minimal. We have $\downarrow e_1 \subseteq B_1 \cap B_2$. Let $e_2 \in B_2$ with $\ell(e_2) = \ell(e_1)$ and $\downarrow e_2 = \downarrow e_1$. We have $e_1 \# e_2$. Hence, $X_{B_1} \times X_{B_2} \subseteq \#$.

Second, we check that $\rho = (E', \leq, \#, \ell, \sigma)$ is a run of \mathcal{A} over ra . We still have $\ell^{-1}(\perp) = \{e_\perp\}$ and $e_\perp \leq e$ for all $e \in E'$. Let $B \in \chi$ and $e \in X_B$. We have $\ell(e) = a$ and $\downarrow e = B$, hence (3) is satisfied. We have $\sigma(e) = e \in X_B \subseteq Q_{W(a)}$. Moreover $H(e) = X_B$, hence $\sigma(H(e)) = X_B \models \delta_a(\bar{\sigma}(B)_{R(a)}) = \delta_a(\bar{\sigma}(\downarrow e)_{R(a)})$ and (5) is satisfied. It remains to check that (4) holds. Let $C' \in \mathcal{C}_{\max}(\rho)$. If $C' \subseteq E$ then $C' \in \mathcal{C}_{\max}(\rho_r)$. Let $B = \partial_{R(a)}C'$ and $e \in X_B$. The set $C' \cup \{e\}$ is a configuration of ρ that strictly contains C' , a contradiction. Then $C' \setminus E = \{e\}$ is a singleton. Let $B \in \chi$ with $e \in X_B$. We have $B = \downarrow e \subseteq C'$. Let $C = C' \cap E$ and $C'' \in \mathcal{C}_{\max}(\rho_r)$ with $C \subseteq C''$. The set $C'' \cup \{e\}$ is a configuration of ρ and since C' is maximal, we deduce that $C = C''$. Therefore $(C, \leq, \ell) \sim \perp \cdot r$. Since

$\Downarrow e = B$ and $(B, \leq, \ell) \sim \perp \cdot \partial_{R(a)} r$ we get $(C', \leq, \ell) \sim \perp \cdot r \cdot a$ and (4) holds. We have proved that ρ' is a run over $r \cdot a$.

Let $s \in (\perp, q^0)K^*$ be an f -play such that $C \approx s$. Let $\xi = (X_B, \ddot{a})_{i \in W(a)}$ and $q = \sigma(e) = e \in X_B$. We have $\xi \in f(\partial_{R(a)} s, \dot{a})$ and $(\xi, q) \in T_{\dot{a}}$. Hence, $s' = s \cdot (\dot{a}, \xi)(\ddot{a}, q) \in (\perp, q^0)K^*$ is an f -play and we have $C' \approx s'$. \square

Lemma 13. *If team 0 has a WDS in the game G_t then t is accepted by \mathcal{A} .*

Proof. Let f be a WDS for team 0 in the game G_t and let $\rho = (E, \leq, \#, \ell, \sigma)$ be a run of \mathcal{A} on t given by Lemma 12. We have to show that ρ is accepting. Let $C \in \mathcal{C}_{\max}(\rho)$ and $s = (\perp, q^0)s'$ be an f -play with $s' \in K^*$ and $C \approx s$. We have $\Pi(s') = t$ hence by Lemma 11 we deduce that s is f -maximal.

Since f is a WDS, the f -maximal f -play s must be in \mathcal{W}_t . Since $s' \in K^*$ there is no $s'' \in \Gamma_1^{'+}$ such that $s's'' \in K^*$. Therefore, $s \in \mathcal{W}_t^1$ and $\bar{\sigma}(s) \in \mathcal{F}$. Using $C \approx s$ we get $\bar{\sigma}(C) = \bar{\sigma}(s)$ and $\bar{\sigma}(C) \in \mathcal{F}$. \square

Now, we show the converse of Lemma 13. We start with an accepting run $\rho = (E, \leq, \#, \ell, \sigma)$ of \mathcal{A} over t . Without loss of generality, we may assume that ρ is reduced. For $\dot{a} \in \Gamma_0$ and $s \in \mathbb{M}(\Gamma', D')$ with $\partial_{R(a)} s = s$ we let $f(s, \dot{a}) = \emptyset$ if $s \notin (\perp, q^0)K^*$ and $f(s, \dot{a}) = \{\nu(e) \mid e \in E \ \& \ \ell(e) = a \ \& \ \Downarrow e \approx s\}$ otherwise. We show that f is a WDS for team 0 in G_t with several lemmas.

Lemma 14. *The mapping f is a deterministic DS for team 0 in the game G_t .*

Proof. Let $\dot{a} \in \Gamma_0$, $s \in (\perp, q^0)K^*$ and $e \in E$ with $\ell(e) = a$ and $\Downarrow e \approx s$. We have $\bar{\sigma}(s) = \bar{\sigma}(\Downarrow e)$. Since ρ is a run, we have $\sigma(H(e)) \models \delta_a(\bar{\sigma}(\Downarrow e)_{R(a)})$. Therefore, $(\bar{\sigma}(s)_{R(a)}, \nu(e)) \in T_{\dot{a}}$ and f is a DS.

For $i = 1, 2$, let $e_i \in E$ with $\ell(e_i) = a$ and $\Downarrow e_i \approx s$. Then $(\Downarrow e_1, \leq, \ell, \sigma)$ and $(\Downarrow e_2, \leq, \ell, \sigma)$ are isomorphic and by Lemma 9 we get $\Downarrow e_1 = \Downarrow e_2$. Therefore, $H(e_1) = H(e_2)$ and $\nu(e_1) = \nu(e_2)$: the DS f is deterministic. \square

Lemma 15. *For each f -play $s \in (\perp, q^0)K^*$, there exists a configuration C of ρ such that $C \approx s$.*

Proof. By induction on $|s|$. If $|s| = 1$ then we take $C = \{e_{\perp}\}$. Assume that $s = (V, \leq, \ell, \sigma) \in (\perp, q^0)K^+$. Let $z \in V$ be maximal and let $y < z$ be an immediate predecessor of z . Since $s \in (\perp, q^0)K^+$ we have $\ell(y) = \dot{a}$, $\sigma(y) = \xi = (X, \ddot{a})_{i \in W(a)}$ for some $X \in Q_{W(a)}$, $\ell(z) = \ddot{a}$ and $\sigma(z) = q \in X$.

Then, $s' = (V \setminus \{y, z\}, \leq, \ell, \sigma) \in (\perp, q^0)K^*$ is an f -play and, by induction, we find a configuration C' of ρ such that $C' \approx s'$. Since s is an f -play, we have $\sigma(y) \in f(\partial_{R(a)} s', \dot{a})$. Let $e \in E$ such that $\ell(e) = a$, $\sigma(y) = \nu(e)$ and $\Downarrow e \approx \partial_{R(a)} s'$. From $C' \approx s'$ we deduce $\partial_{R(a)} C' \approx \partial_{R(a)} s'$. Therefore, $(\Downarrow e, \leq, \ell, \sigma)$ is isomorphic to $(\partial_{R(a)} C', \leq, \ell, \sigma)$. By Lemma 9 we get $\Downarrow e = \partial_{R(a)} C'$.

Then, $C = C' \cup \{e\}$ is a configuration of ρ . If we extend $\cdot : C' \rightarrow V \setminus \{y, z\}$ by $\ddot{e} = z$ then we have $\dot{e} = y$ and we get $C \approx s$. \square

Lemma 16. *For each f -play s_1 , there exists $s_2 \in \Gamma_1^{'+}$ such that $s_1 s_2 = (\perp, q^0)s$ with $s \in K^*$ and $\Pi(s) \leq t$.*

Proof. Let $s_1 = (V_1, \leq, \ell, \sigma)$ be an f -play and $s_0 = (V_0, \leq, \ell, \sigma)$ be the largest prefix of s_1 in $(\perp, q^0)K^*$. We use an induction on $s_0^{-1}s_1$. If $s_1 = s_0$ then by Lemma 15 we find a configuration C of ρ such that $C \approx s_1$. Then, $s_1 = (\perp, q^0)s$ ($s_2 = \varepsilon$) and $\perp \cdot \Pi(s) = (C, \leq, \ell) \leq \perp \cdot t$ by (4). Therefore, $\Pi(s) \leq t$.

Now, assume that $V_1 \setminus V_0 \neq \emptyset$ and let x_1, \dots, x_p be the minimal events in $V_1 \setminus V_0$. Since $s_0 \in (\perp, q^0)K^*$ we have $\bar{\sigma}(s_0) \in Q$ and only moves of team 0 are enabled at s_0 . Therefore, $\ell(x_i) \in \Gamma_0$ for each $1 \leq i \leq p$.

Assume that $V_0 \cup \{x_1, \dots, x_p\} \subsetneq V_1$ and let y be a minimal event in $V_1 \setminus (V_0 \cup \{x_1, \dots, x_p\})$. Then, $x_n \prec y$ for some $1 \leq n \leq p$. We have $\ell(y) \text{D} \ell(x_n)$ hence there exists $i \in R(\ell(y)) \cap W(\ell(x_n))$. We have seen that $\ell(x_n) = \dot{a} \in \Gamma_0$, hence $\sigma(x_n) = (X, \ddot{a})_{i \in W(a)}$ for some $X \in Q_{W(a)}$. Therefore, $\bar{\sigma}(\downarrow y)_i = \sigma(x_n)_i = (X, \ddot{a})$. Hence, $\ell(y) = \ddot{a} \in \Gamma_1$ and $\sigma(y) \in X$. Then, $s_0(\ell(x_n), \sigma(x_n))(\ell(y), \sigma(y)) \in (\perp, q^0)K^*$ is a prefix of s_1 , which is a contradiction with the maximality of s_0 .

Therefore, $V_1 = V_0 \cup \{x_1, \dots, x_p\}$. Let $\dot{a} = \ell(x_1)$, $\sigma(x_1) = (X, \ddot{a})_{i \in W(a)}$ and $q \in X$. We have $\bar{\sigma}(s_1)_i = \sigma(x_1)_i = (X, \ddot{a})$ for all $i \in W(a)$. Hence, $(\bar{\sigma}(s_1)_{R(a)}, q) \in T_{\ddot{a}}$ and $s'_1 = s_1(\ddot{a}, q)$ is an f -play. Note that (\ddot{a}, q) is independent from $(\ell(x_i), \sigma(x_i))$ for $1 < i \leq p$ since the x_i 's are pairwise concurrent events in s_1 . Hence $s'_0 = s_0(\dot{a}, \sigma(x_1))(\ddot{a}, q)$ is the largest prefix of s'_1 in $(\perp, q^0)K^*$. Then, $|s_0^{-1}s_1| < |s'_0^{-1}s'_1|$ and by induction, we find $s'_2 \in \Gamma_1'^*$ such that $s'_1 s'_2 = (\perp, q^0)s$ with $s \in K^*$ and $\Pi(s) \leq t$. Let $s_2 = (\ddot{a}, q)s'_2$. We have $s_1 s_2 = s'_1 s'_2$ which concludes the proof. \square

Lemma 17. *The mapping f is a WDS for team 0 in the game G_t .*

Proof. Let s_1 be an f -maximal f -play with $s_1 \notin (\perp, q^0)K^*$. By Lemma 16, there exists $s_2 \in \Gamma_1'^+$ such that $s_1 s_2 = (\perp, q^0)s$ with $s \in K^*$ and $\Pi(s) \leq t$. Therefore, $s_1 \in \mathcal{W}_t$.

Let $s = (\perp, q^0)s'$ be an f -maximal f -play with $s' \in K^*$. By Lemma 15 we find a configuration C of ρ such that $C \approx s$. If C is not maximal, let $e \notin C$ with $C \cup \{e\}$ configuration of ρ and let $a = \ell(e)$. We have $\downarrow e = \partial_{R(a)}C \approx \partial_{R(a)}s$ hence $\nu(e) \in f(\partial_{R(a)}s, \dot{a})$ which is a contradiction with the fact that s is f -maximal. Therefore, $C \in \mathcal{C}_{\max}(\rho)$ and $(C, \leq, \ell) \sim t$ by (4). Using $C \approx s$ we deduce that $\Pi(s') = t$. Now, since ρ is accepting, we have $\bar{\sigma}(C) \in \mathcal{F}$ and using $C \approx s$ we deduce that $\bar{\sigma}(s) = \bar{\sigma}(C) \in \mathcal{F}$. Therefore, $s \in \mathcal{W}_t$. \square

Acknowledgment We thank Madhavan Mukund for fruitful discussions.

References

1. R. Cori, Y. Métivier, and W. Zielonka. Asynchronous mappings and asynchronous cellular automata. *Inform. and Comput.*, 106:159–202, 1993.
2. V. Diekert and Y. Métivier. *Handbook of Formal languages*, volume 3, chapter Partial Commutations and Traces, pages 457–533. Springer, 1997.
3. V. Diekert and G. Rozenberg, editors. *The Book of Traces*. World Scientific, 1995.
4. O. Kupferman and M. Y. Vardi. Synthesizing distributed systems. In *Proc. of the LICS '01*. Computer Society Press, 2001.

5. P. Madhusudan and P. S. Thiagarajan. Distributed controller synthesis for local specifications. In *Proc. of the ICALP '01*, volume 2076 of *Lect. Notes Comp. Sci.*, pages 396–407. Springer, 2001.
6. P. Madhusudan and P. S. Thiagarajan. A decidable class of asynchronous distributed controllers. In *Proc. of the CONCUR '02*, volume 2421 of *Lect. Notes Comp. Sci.*, pages 145–160. Springer, 2002.
7. S. Mohalik and I. Walukiewicz. Distributed games. In *FSTTCS '03*, *Lect. Notes Comp. Sci.* Springer, 2003.
8. M. Nielsen, G. Plotkin, and G. Winskel. Petri nets, event structures and domains. *Theoret. Comput. Sci.*, 13(1):85–108, 1981.
9. J. Pin and D. Perrin. *Infinite words. Automata, Semigroups, Logic and Games*. Elsevier, To appear.
10. A. Pnueli and R. Rosner. Distributed reactive systems are hard to synthesize. In *Proc. of the 31th IEEE Symp. FOCS*, pages 746–757, 1990.
11. P. Ramadge and W. Wonham. The control of discrete event systems. In *Proceedings of the IEEE*, volume 77, pages 81–98, 1989.
12. W. Thomas. Infinite games and verification. In *Proc. of CAV'02*, volume 2404 of *Lect. Notes Comp. Sci.*, pages 58–64. Springer, 2002.
13. J. Vöge and M. Jurdziński. A discrete strategy improvement algorithm for solving parity games. In *Proc. of CAV '00*, volume 1855 of *Lect. Notes Comp. Sci.*, pages 202–215. Springer, 2000.
14. W. Zielonka. Asynchronous automata. In G. Rozenberg and V. Diekert, editors, *Book of Traces*, pages 175–217. World Scientific, Singapore, 1995.
15. W. Zielonka. Infinite games on finitely coloured graphs with applications to automata on infinite trees. *Theoretical Computer Science*, 200(1–2):135–183, 1998.