

Distributed games and distributed control for asynchronous systems

P. Gastin, B. Lerman, M. Zeitoun

LIAFA, CNRS & Univ. Paris 7

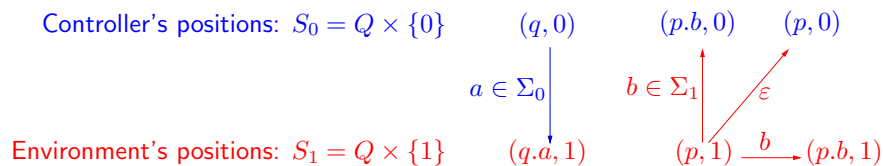
A control synthesis problem — sequential case

- Finite open transition system \mathcal{A} over alphabet



- Set of **good** behaviors: regular language \mathcal{G} .
- Find a controller \mathcal{C} (finite automaton) such that
 - \mathcal{C} selects actions of Σ_0 but cannot disable actions from Σ_1 .
 - All Σ_0 -maximal behaviors of $\mathcal{A} \times \mathcal{C}$ should be in \mathcal{G} .

A game associated with the control synthesis problem



Play: finite or infinite sequence of moves starting from $(s_{in}, 1)$.

Player 0 (the controller) wins a play iff its label is in \mathcal{G} .

A strategy f for player 0 tells him how to play on $S_0 = Q \times \{0\}$.

A strategy f for player 0 is winning if all f -maximal f -plays are winning.

Strategies

Memoryless: depends only on the current state.

$$f : S_0 \times \Sigma_0 \rightarrow S_1 \cup \text{Stop}$$

Perfect memory: depends on the whole prefix played so far.

$$f : S^*.S_0 \times \Sigma_0 \rightarrow S_1 \cup \text{Stop}$$

Facts

- Player 0 has a winning strategy in the game iff there exists a controller for \mathcal{A} .
- Player 0 has a winning strategy in the game if and only if he has a **finite memory** winning strategy.
- One can decide whether player 0 has a winning strategy in that game.

Summary

Games well suited to solve the control synthesis problem.

- System = game graph.
- Controller = a player.
- Environment = opponent.
- Good behavior = winning condition.
- Finding a controller = finding a winning strategy.

Control synthesis problem — distributed case

- Communicating systems \mathcal{A}_i over a given architecture, on several locations.



- Set of good behaviors \mathcal{G} .

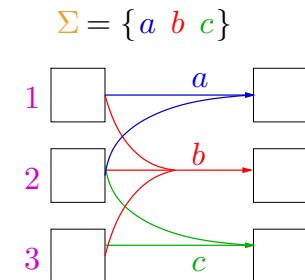
Goal Build local controllers \mathcal{C}_i , synchronized with \mathcal{A}_i , so that

- \mathcal{C}_i selects actions of $\Sigma_{0,i}$ but cannot disable actions from $\Sigma_{1,i}$.
- The behavior of the overall system $(\mathcal{A}_i \times \mathcal{C}_i)_i$ is in \mathcal{G} (regardless of the moves of the environment.)

Distributed games

Distributed architectures (Zielonka)

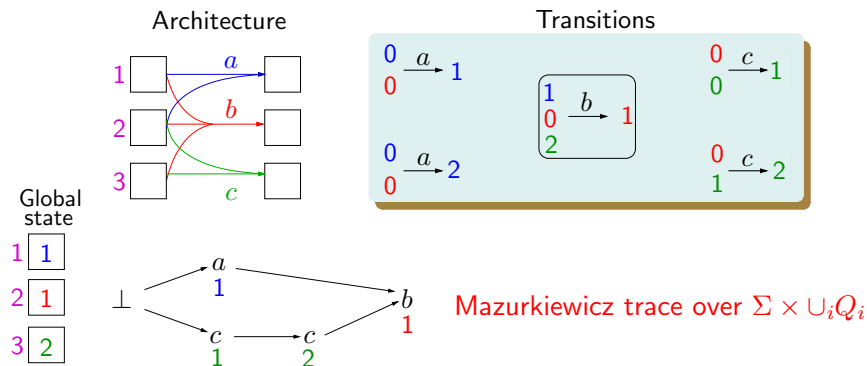
- Σ : finite set of actions or players
- \mathcal{P} : finite set of processes (memory cells)
- $R : \Sigma \rightarrow 2^{\mathcal{P}}$ assigns to each $a \in \Sigma$ its read domain $R(a)$
- $W : \Sigma \rightarrow \mathcal{P}$ assigns to each $a \in \Sigma$ its write domain $W(a)$.
In this talk: $|W(a)| = 1$.
- Action a changes the state of process $W(a)$ according to states read in $R(a)$.



Asynchronous distributed games (1)

$\langle (Q_i)_{i \in \mathcal{P}}, (T_a)_{a \in \Sigma}, q^0 \rangle$ distributed automaton over $(\Sigma, \mathcal{P}, R, W)$

- $\forall i \in \mathcal{P}, Q_i$ is the set of local states for process i
- $\forall a \in \Sigma, T_a \subseteq Q_{R(a)} \times Q_{W(a)}$, possible local moves of action a
- $q^0 \in Q = \prod_{i \in \mathcal{P}} Q_i$ is the global starting position



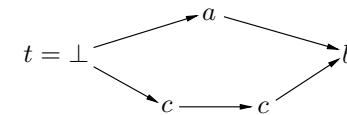
Mazurkiewicz traces

Dependence alphabet (Σ, D) , D reflexive & symmetric. $(\Sigma, D) = \perp \begin{matrix} \swarrow a \\ \longrightarrow b \\ \searrow c \end{matrix}$

Actions labeled by a, b with $a D b$ share resources and cannot be concurrent.

Trace $t = (V, \leq, \lambda)$ st. $\lambda : V \rightarrow \Sigma$, and the labeling λ constrains the partial order \leq : $x < y \Rightarrow \lambda(x) D \lambda(y)$ $\lambda(x) D \lambda(y) \Rightarrow x \leq y$ or $y \leq x$

Traces enjoy several properties (eg uniqueness of a trace from a linearization)



Technical restriction

$$\forall a \in \Sigma, \quad \emptyset \neq W(a) \subseteq R(a)$$

$$\forall a, b \in \Sigma, \quad R(a) \cap W(b) = \emptyset \iff R(b) \cap W(a) = \emptyset \quad (\S)$$

Dependence relation over Σ : $D = \{(a, b) \mid R(a) \cap W(b) \neq \emptyset\}$

Asynchronous distributed games (2)

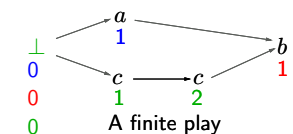
- $\Sigma = \Sigma_0 \uplus \Sigma_1$: partition of the set of players (actions) in teams 0 and 1. Players of team 0 cooperate together against team 1.
- Winning condition: set of finite or infinite traces $\mathcal{W} \subseteq \mathbb{R}(\Sigma', D)$. Team 0 wins plays of \mathcal{W} and loses plays of $\mathbb{R}(\Sigma', D) \setminus \mathcal{W}$.

Asynchronous distributed games (3)

- $\Sigma = \Sigma_0 \uplus \Sigma_1$: partition of the set of players (actions) in teams 0 and 1. Players of team 0 cooperate together against team 1.
- Each player only has a partial view of the global history.

$$\Sigma' = \{(a, p) \mid a \in \Sigma, p \in Q_{W(a)}\} \cup \{(\perp, q^0)\}, \quad (a, p) D (b, q) \iff a D b$$

- “Positions” of the game $V \subseteq \mathbb{M}(\Sigma', D)$.
- Play: finite or infinite trace of $\mathbb{R}(\Sigma', D)$.
- Rules of the game for player a given by T_a .



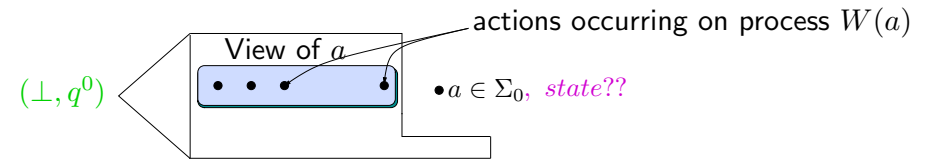
- Winning condition: set of finite or infinite traces $\mathcal{W} \subseteq \mathbb{R}(\Sigma', D)$. Team 0 wins plays of \mathcal{W} and loses plays of $\mathbb{R}(\Sigma', D) \setminus \mathcal{W}$.
- Remarks Teams 0 and 1 might be simultaneously enabled at $v \in V$.

One can recover the global state from a distributed play.

Distributed strategies

Distributed strategies with local memory

Recall that a **distributed play** is a rooted trace of $\mathbb{R}(\Sigma', D)$.



$$f : \text{View of } a, a \mapsto Q_{W(a)} \cup \{\text{Stop}\}$$

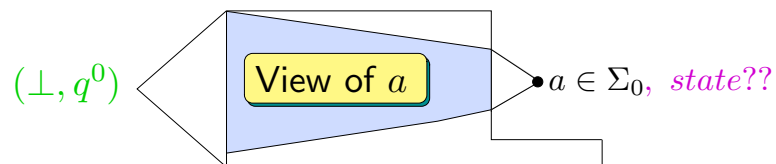
The strategy for a does **not** use the **full history**, only a 's local view.

Used by [Thiagarajan-Madhusudan 02,03] and by [Bernet-Janin-Walukiewicz 04].

Distributed strategies with causal memory

Why causal memory?

Recall that a **distributed play** is a rooted trace of $\mathbb{R}(\Sigma', D)$.



$$f : s, a \mapsto Q_{W(a)} \cup \{\text{STOP}\}$$

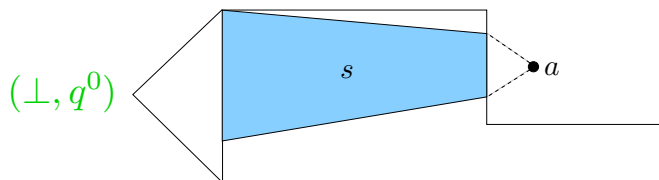
All other causal distributed strategies are abstractions of this perfect memory distributed strategy.

- Modeling
 - Enforcing the environment to forward information may be realistic.
- Computability
 - Causal memory is the largest possible memory in a distributed setting.
 - The existence of a **global** winning strategy is a **necessary** condition for the existence of a winning **distributed** strategy.
 - The existence of a winning strategy with **causal memory** is a **necessary** condition for the existence of a winning distributed strategy.

Distributed strategies: maximality condition

Let f be a distributed strategy.

An f -play is f -maximal if no Σ_0 -move compatible with f can be performed.



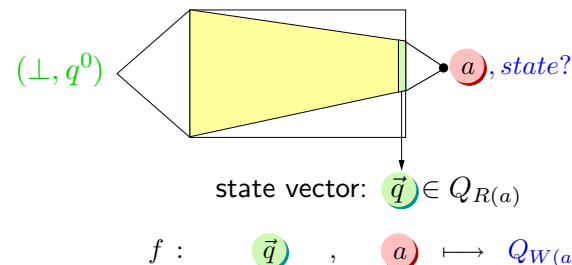
$$a \in \Sigma_0 \implies f(s, a) = \text{STOP}$$

The winning condition only talks about f -maximal plays:

A strategy f is **winning** if all f -maximal f -plays are in \mathcal{W} .

Memoryless distributed strategies

Coarse abstraction of the causal perfect memory:



Finer abstractions of perfect memory \rightarrow distributed memories μ and strategies.

Proposition For a distributed game G and a distributed memory μ , one can build a game G^μ such that team 0 has a WDS in G with memory μ iff it has a memoryless WDS in G^μ .

Proof $G^\mu = G \times \mu$.

Deciding distributed strategies

Proposition Deciding whether team 0 has a **causal** distributed (memoryless) WS is **undecidable** for rational winning conditions.

Theorem Deciding whether team 0 has a winning **local** distributed strategy is **undecidable** [BJW04] even:

- for reachability or safety winning conditions.
- with 3 players.

With causal memory, this undecidability result (with 3 players) does not hold.

Theorem Deciding whether team 0 has a winning **causal** distributed strategy is **decidable** for all reachability conditions.

Existence of distributed strategies: undecidability

Proposition Deciding whether team 0 has a WDS/WMDS is undecidable (even for rational winning conditions).

Proof. $\text{Rat}(\Sigma, D)$: trace languages of the form $[R]$, $R \in \text{Rat}(\Sigma^*)$.

Undecidable: whether $\mathcal{L} \in \text{Rat}(A^* \times B^*)$ equals to $A^* \times B^*$.

$$(|A|, |B| \geq 2)$$

Let $G_{\mathcal{L}}$ be the following game:

- Team 0 has no players
- Team 1 has players $a_i \in A$ and $b_j \in B$ with $D = A^2 \uplus B^2$.
- $\mathcal{W} = \mathcal{L} \cup \{\text{infinite traces}\}$.

Team 0 has a (memoryless/perfect memory/arbitrary) WDS in $G_{\mathcal{L}}$

$$\iff \mathcal{L} = A^* \times B^*$$

Distributed games [MW03]

Distributed games studied for long: [PR89], [MT01,02], [MW03]...

Built from local games $G_i = \langle P_i, E_i, Tr_i, q_i^0 \rangle$, $Tr_i \subseteq (P_i \times E_i)$.

Positions

- ▷ Players: $P = \prod_i (P_i \cup E_i) \setminus E$.
- ▷ Environment: $E = \prod_i E_i$.

Transitions: $Tr = Tr_p \uplus Tr_e$.

- ▷ Players: **Cartesian product:** $Tr_p = (\prod_i (Tr_i \cup \Delta_i)) \cap (P \times E)$ Δ_i : diagonal
- ▷ Environment: **Global**, given by a subset Tr_e of $E \times P$.

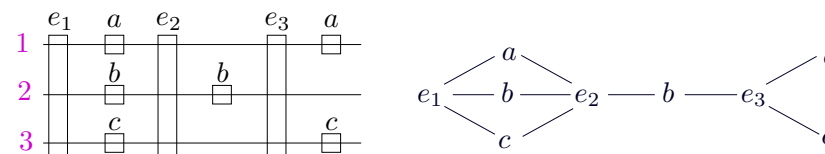
↔ Informations between players are passed by the environment.

Moves of players and of the environment alternate.

Winning condition: $Acc \subseteq (E \cdot P)^\omega$.

Distributed games [MW03]

Plays of these distributed games can be represented by traces.

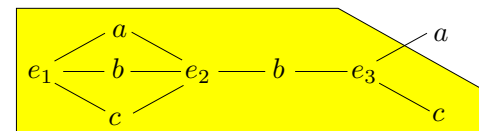


Remark: the alphabet is serie-parallel.

Important: The strategies use a **local** memory.

A player's strategy only depends on the sequence $e_1 p_1 e_2 p_2 \dots$ he observed on **his** memory cell: his view is **local** and not **causal**.

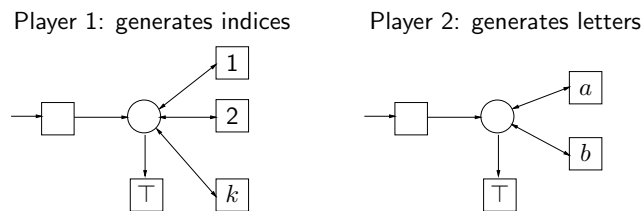
With a causal strategy, a player could detect asynchronous moves.



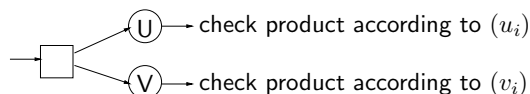
Undecidability of [MW03] distributed games [BJW03]

Theorem Even **accessibility** is undecidable for [MW03] distributed games.

From an instance $u_1, v_1, \dots, u_k, v_k \in \{a, b\}^*$ of PCP, build a 3 players game



Player 3: receives from the environment these sequences of indices and letters and check that the product is correct w.r.t. (u_i) or (v_i) .



Decidability of accessibility in ADG for serie-parallel alphabets

Remarks on accessibility games

Several kinds of accessibility games

- **Global:** reach a global state vs. **Local:** reach a local state.
- **Strict:** reach & stop vs. **Relaxed:** just reach.

- **Distributed games [MW03]**
Determining whether team 0 has a winning strategy is **undecidable** for accessibility games with at least 3 players.
- **Asynchronous distributed games**
Same problem with causal memory is **decidable** for serie-parallel alphabets (all flavors of accessibility conditions).

Proof (1) induction on the alphabet.

(2) transform a WDS in another one with bounded memory.

Proof (1): strategies have finite memory

Proposition If Σ_0 has a winning distributed strategies f on a strict accessibility game, then the set of f -maximal f -plays is finite.

In particular, f has a **finite** memory.

Proof König's Lemma on the tree of f -plays..

- With no bound on the memory, this remark does not yield an algorithm.
- **Aim** compute a **bound** for the memory, depending on $|\Sigma|$ and $|Q|$ only.

Remark For all $M > 0$ there is an accessibility game strict/relaxed global/local in which Σ_0 needs d'at least memory of size M .

Following memories are not enough: :

- ★ Global state of a 's view.
- ★ Finite approximations of a 's view.

Proof (2): from the set of f -plays, recover f

- Let $P(f)$ the set of f -maximal f -plays.
- We know that $P(f)$ is **finite**.
- If t is a play, let $\bar{q}(t)$ be the global state reached by t .
- If P is a set of plays, there is a strategy f such that $P = P(f)$ iff
 - P follows the rules of the game,
 - P contains all possible moves of team 1,
 - Team 0 plays in a distributed & deterministic way in P .

Proof (3): induction principle

Proposition One can compute $S_M : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ st for all $\Sigma, Q, \delta, q_0, F$, if there is a winning distributed strategy f for team 0 in the game $(\Sigma, Q, \delta, q_0, F)$, then there is another such strategy using **less than** $S_M(|\Sigma|, |Q|)$ memory.

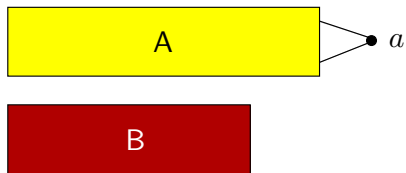
Proof Induction on $|\Sigma|$.

$f \rightsquigarrow P(f) \rightsquigarrow P' = P(f')$ such that f' uses at most $S_M(|\Sigma|, |Q|)$ memory.

For $|\Sigma| = 1$, the game is **sequential**.

Proof (4): case $\Sigma = A \parallel B$

For $\Sigma = A \parallel B$, $\bar{q}(P(f))$ is a direct product $\mathcal{W}_A \times \mathcal{W}_B$



One considers the projections f_A and f_B of f on G_A and G_B .

Induction hypothesis gives f'_A on G_A using less than $S_M(|A|, |Q|)$ memory.

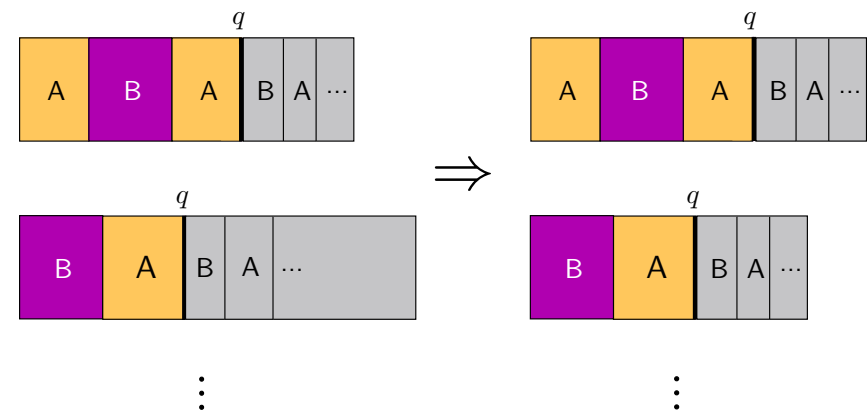
f'_B on G_B using less than $S_M(|B|, |Q|)$ memory.

Then $f' = f'_A \times f'_B$ is winning on G and uses less than:

$$S_M(|A|, |Q|) + S_M(|B|, |Q|) \text{ memory.}$$

Proof (5): case $\Sigma = A \otimes B$

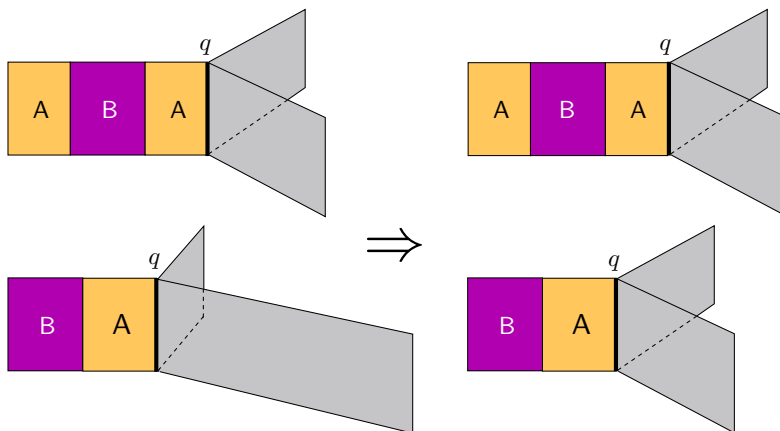
For $\Sigma = A \otimes B$, one transforms $P(f)$ into P' .



Proof (6): case $\Sigma = A \otimes B$ (continued)

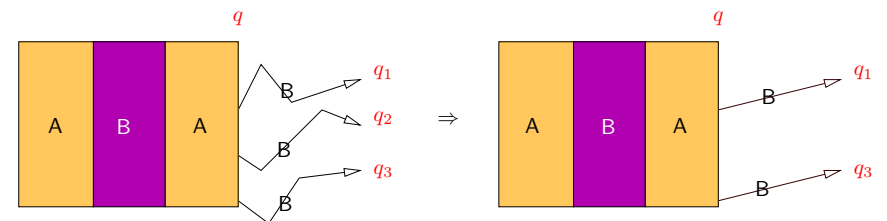
These cuts & pastes done **uniformly** on all suffixes extending a context (A, q) .

One keeps the set of suffixes which is the fastest to win in the worst case.



Proof (7)

- One obtains a new strategy f' such that $P' = P(f')$.
- From f' , consider the small strategies reaching $\{q_i\}$ from A staying in B . Transform them using the induction hypothesis and glue them to build P'' .



The set of plays P'' corresponds to a strategy f'' with memory **less than**:

$$\max(S_M(|A|, |Q|), S_M(|B|, |Q|)) + |Q|.$$

From distributed games to sequential games

From a distributed game to its associated sequential game

Theorem Given a finite distributed game (G, \mathcal{W}) , we can effectively build a finite sequential 2-players game $(\tilde{G}, \tilde{\mathcal{W}})$ st. the following are equivalent:

1. There exists a **memoryless distributed** WS for team 0 in (G, \mathcal{W}) .
2. There exists a memoryless WS for player 0 in $(\tilde{G}, \tilde{\mathcal{W}})$.
3. There exists a WS for player 0 in $(\tilde{G}, \tilde{\mathcal{W}})$.

Naive idea Consider the game on the global transition system.

Main problem The controller has more information than its causal memory.

Solution

- The opponent controls the linearization to be played.
- Using resets moves, he can replay different linearizations for the same play.
- The winning condition $\tilde{\mathcal{W}}$ makes sure that the strategy followed by the controller is indeed distributed.

Sequentialized game \tilde{G}

Player 1 wins if he shows that player 0

- does not reach \mathcal{W} , or
- does not follow a **distributed** strategy.

Player 1 can

- decide which actions are used and in which order.
- investigate all possible linearizations of distributed plays with **reset moves**

Positions: $Z = \underbrace{Q \times \Sigma_0}_{Z_0} \uplus \underbrace{Q \times (\Sigma_1 \cup \{0, 1, 2\})}_{Z_1}$, player i plays on Z_i .

- $(q, a) \in Z$
- q stores the global state;
 - $a \in \Sigma_i$: player i has to play a . Player 0 may **refuse** the move
 - $a = 0$: a reset move has just be performed by player 1.
 - $a = 1$: coming from (p, b) , the b -move has been performed
 - $a = 2$: player 0 refused the previous move.

Sequentialized game: example

2 processes 1, 2. $\Sigma_i = \{a_i, b_i\}$, $R(a_i) = W(a_i) = 1$ and $R(b_i) = W(b_i) = 2$.

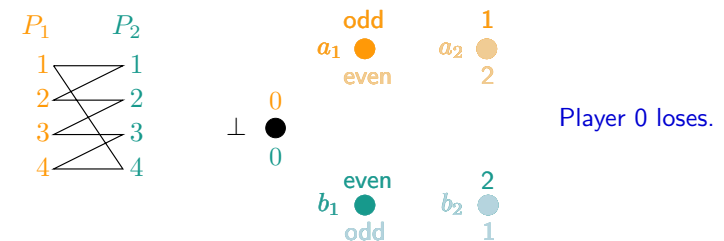
Σ_1 -transitions: $q^0 \xrightarrow{a_1} \text{odd} \vee \text{even}$ and $q^0 \xrightarrow{b_1} \text{odd} \vee \text{even}$

Σ_0 -transitions: $\text{odd} \xrightarrow{a_0} 1 \vee 3$, $\text{even} \xrightarrow{a_0} 2 \vee 4$

$\text{odd} \xrightarrow{b_0} 1 \vee 3$, $\text{even} \xrightarrow{b_0} 2 \vee 4$

Plays will have the following shape: (\perp, q^0)

$\mathcal{W} = \{(i, j) \mid i - j \equiv 0 \text{ or } 1 \pmod{4}\} \cup \text{Shape}_1$



Conclusion and further work

- ★ In asynchronous distributed games, existence of strategies with causal memory is a necessary condition for the existence of winning strategies.
- ★ This is decidable for accessibility winning conditions for SP alphabets.

- ★ More complicated winning conditions (Büchi, parity, etc).
- ★ General dependence alphabets.
- ★ How much memory is needed? Problem complexity?
- ★ Find a “real” algorithm.

References

- [1] Zielonka. Notes on finite asynchronous automata. [RAIRO ITA](#), 1987.
- [2] Zielonka. Asynchronous automata. [Book of Traces](#), 1995.
- [3] Pnueli & Rosner. Distributed reactive systems are hard to synthesize. [FOCS'90](#)
- [4] Madhusudan & Thiagarajan. Distributed controller synthesis for local specifications. [ICALP'01](#)
- [5] Madhusudan & Thiagarajan. Effective strategies for asynchronous distributed control. [CONCUR'02](#)
- [6] Mohalik & Walukiewicz. Distributed Games. [FSTTCS'03](#).
- [7] Bednarczyk. Hereditary history preserving bisimulations. TR, Gdansk, 1991.
- [8] Nielsen & Clausen. Games and logics for a noninterleaving bisimulation. [Nordic J. Comput.](#) 1995.
- [9] Jurdziński & Nielsen. Hereditary history preserving bisimilarity is undecidable. [STACS'00](#).
- [10] Mukund. Hereditary history preserving bisimulation is decidable for trace-labeled systems. [FSTTCS'02](#).
- [11] Ramadge & Wonham. The control of discrete event systems. [Proc IEEE](#), 1989.