

Asynchronous distributed games and asynchronous alternating automata

M. Zeitoun, joint work with P. Gastin and B. Lerman

LIAFA, CNRS & Univ. Paris 7

Motivations & contributions of this work

- Framework for [asynchronous](#) multi-players distributed games.
- [Communication](#) to cooperate: 2 teams of players with shared memory.
Some memory locations can be read by some players of both teams.

Motivations & contributions of this work

- Framework for asynchronous multi-players distributed games.
- Communication to cooperate: 2 teams of players with shared memory.
Some memory locations can be read by some players of both teams.
- Definition of **asynchronous distributed games (ADG)**
- Relate ADG and sequential games + back moves

Motivations & contributions of this work

- Framework for asynchronous multi-players distributed games.
- Communication to cooperate: 2 teams of players with shared memory. Some memory locations can be read by some players of both teams.
- Definition of **asynchronous distributed games (ADG)**
- Relate ADG and sequential games + back moves
- Definition of **asynchronous alternating automata (AAA)**
- Natural definition of **runs** by prime event-structures
- Link acceptance in **AAA** and existence of winning strategies in **ADG**

Distributed architectures

$(\Sigma, \mathcal{P}, R, W)$:

- 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 $R(a)$
- $W : \Sigma \rightarrow 2^{\mathcal{P}}$ assigns to each $a \in \Sigma$ its write domain $W(a)$

Distributed architectures

$(\Sigma, \mathcal{P}, R, W)$:

- 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 $R(a)$
- $W : \Sigma \rightarrow 2^{\mathcal{P}}$ assigns to each $a \in \Sigma$ its write domain $W(a)$

$$\Sigma = \{a \ b\}$$

Distributed architectures

$(\Sigma, \mathcal{P}, R, W)$:

- 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 $R(a)$
- $W : \Sigma \rightarrow 2^{\mathcal{P}}$ assigns to each $a \in \Sigma$ its write domain $W(a)$

$$\Sigma = \{a \ b\}$$

1

2

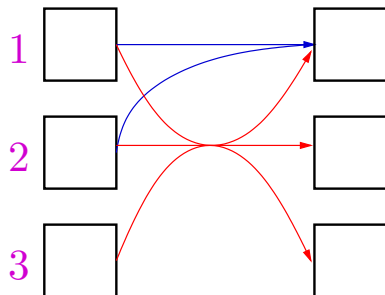
3

Distributed architectures

$(\Sigma, \mathcal{P}, R, W)$:

- 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 $R(a)$
- $W : \Sigma \rightarrow 2^{\mathcal{P}}$ assigns to each $a \in \Sigma$ its write domain $W(a)$

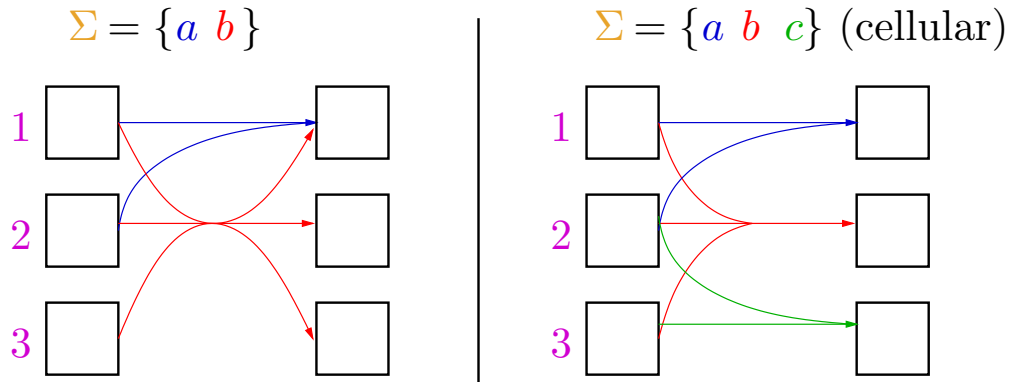
$$\Sigma = \{a \ b\}$$



Distributed architectures

$(\Sigma, \mathcal{P}, R, W)$:

- 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 $R(a)$
- $W : \Sigma \rightarrow 2^{\mathcal{P}}$ assigns to each $a \in \Sigma$ its write domain $W(a)$



Asynchronous distributed systems

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

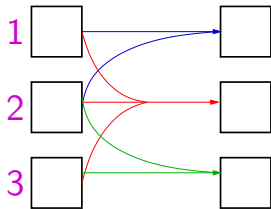
- $\forall i \in \mathcal{P}$, Q_i is the set of local states for process i
- $\forall a \in \Sigma$, $T_a \subseteq Q_{R(a)} \times Q_{W(a)}$, possible local moves of action a
- $q^0 \in Q = \prod_{i \in \mathcal{P}} Q_i$ is the global starting position of G

Asynchronous distributed systems

$\langle (Q_i)_{i \in \mathcal{P}}, (T_a)_{a \in \Sigma}, q^0 \rangle$ asynchronous distributed system 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 of G

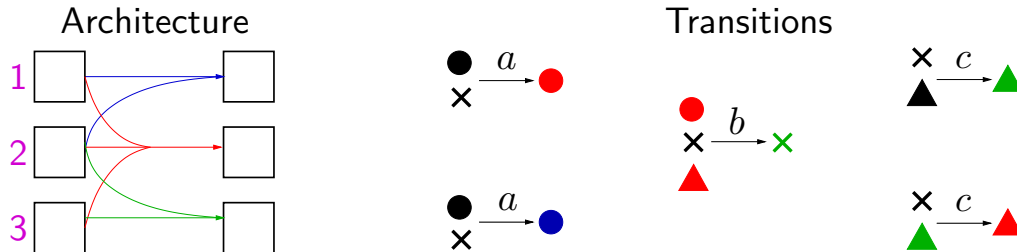
Architecture



Asynchronous distributed systems

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

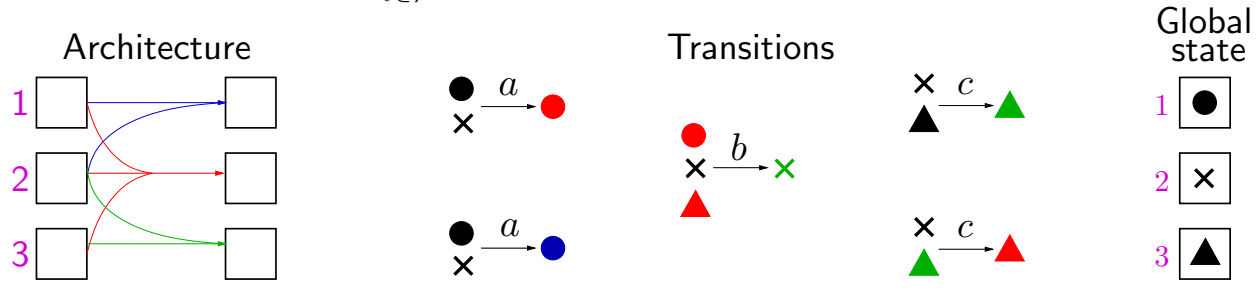
- $\forall i \in \mathcal{P}$, Q_i is the set of local states for process i
- $\forall a \in \Sigma$, $T_a \subseteq Q_{R(a)} \times Q_{W(a)}$, possible local moves of action a
- $q^0 \in Q = \prod_{i \in \mathcal{P}} Q_i$ is the global starting position of G



Asynchronous distributed systems

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

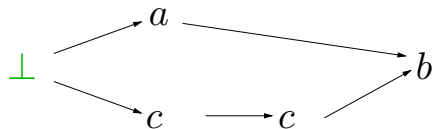
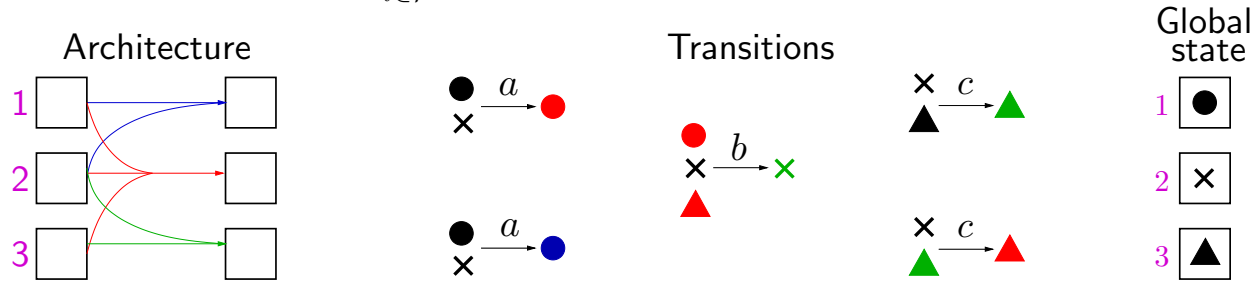
- $\forall i \in \mathcal{P}$, Q_i is the set of local states for process i
- $\forall a \in \Sigma$, $T_a \subseteq Q_{R(a)} \times Q_{W(a)}$, possible local moves of action a
- $q^0 \in Q = \prod_{i \in \mathcal{P}} Q_i$ is the global starting position of G



Asynchronous distributed systems

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

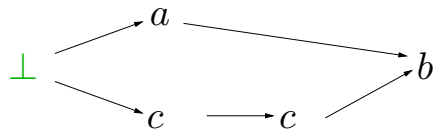
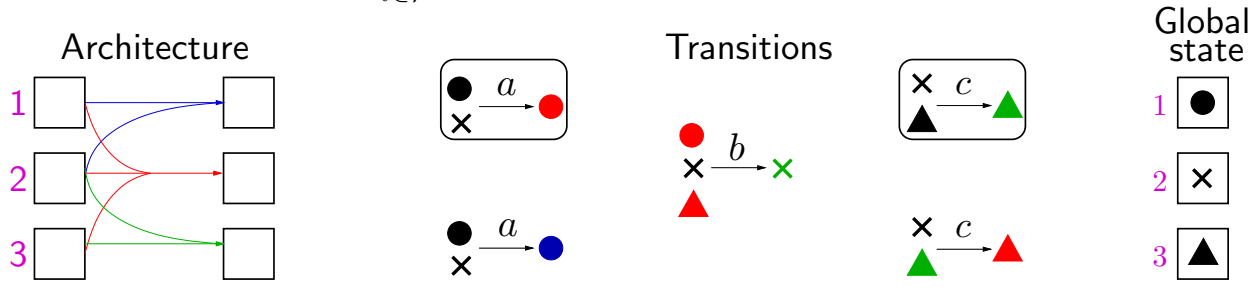
- $\forall i \in \mathcal{P}$, Q_i is the set of local states for process i
- $\forall a \in \Sigma$, $T_a \subseteq Q_{R(a)} \times Q_{W(a)}$, possible local moves of action a
- $q^0 \in Q = \prod_{i \in \mathcal{P}} Q_i$ is the global starting position of G



Asynchronous distributed systems

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

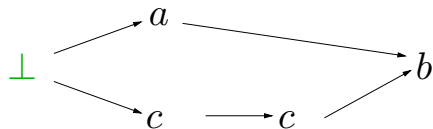
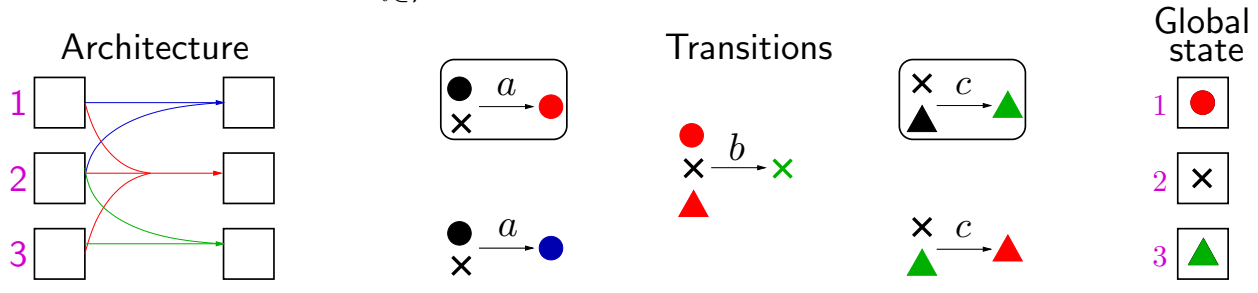
- $\forall i \in \mathcal{P}$, Q_i is the set of local states for process i
- $\forall a \in \Sigma$, $T_a \subseteq Q_{R(a)} \times Q_{W(a)}$, possible local moves of action a
- $q^0 \in Q = \prod_{i \in \mathcal{P}} Q_i$ is the global starting position of G



Asynchronous distributed systems

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

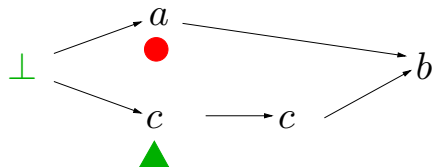
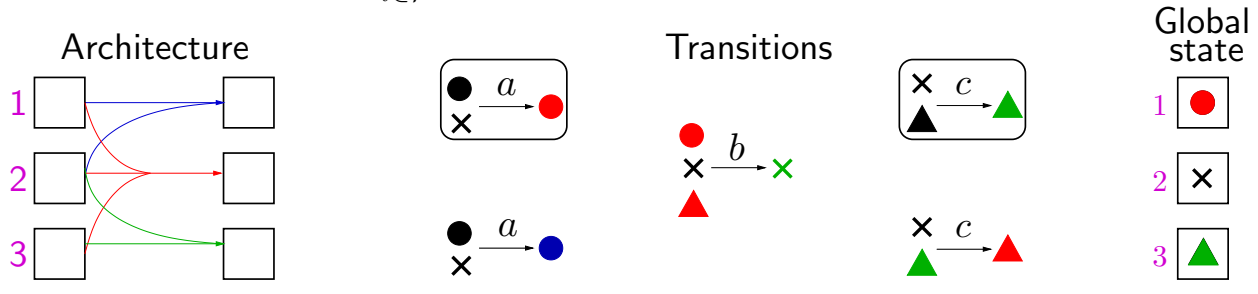
- $\forall i \in \mathcal{P}$, Q_i is the set of local states for process i
- $\forall a \in \Sigma$, $T_a \subseteq Q_{R(a)} \times Q_{W(a)}$, possible local moves of action a
- $q^0 \in Q = \prod_{i \in \mathcal{P}} Q_i$ is the global starting position of G



Asynchronous distributed systems

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

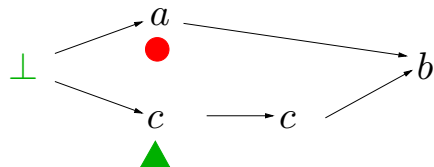
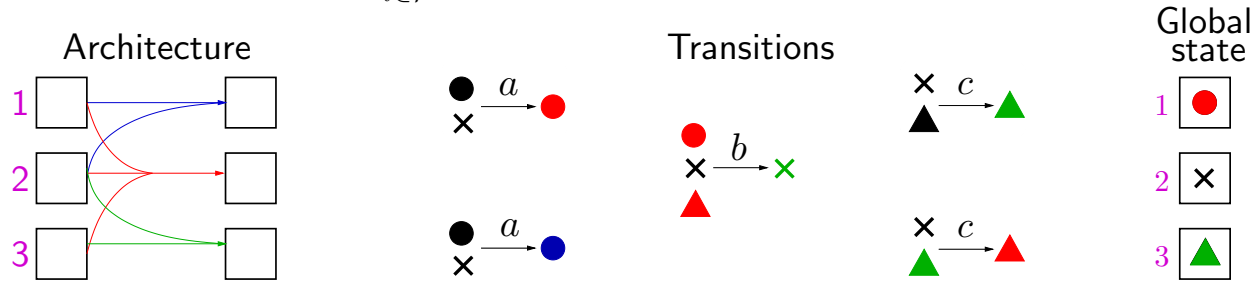
- $\forall i \in \mathcal{P}$, Q_i is the set of local states for process i
- $\forall a \in \Sigma$, $T_a \subseteq Q_{R(a)} \times Q_{W(a)}$, possible local moves of action a
- $q^0 \in Q = \prod_{i \in \mathcal{P}} Q_i$ is the global starting position of G



Asynchronous distributed systems

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

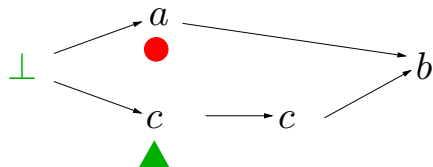
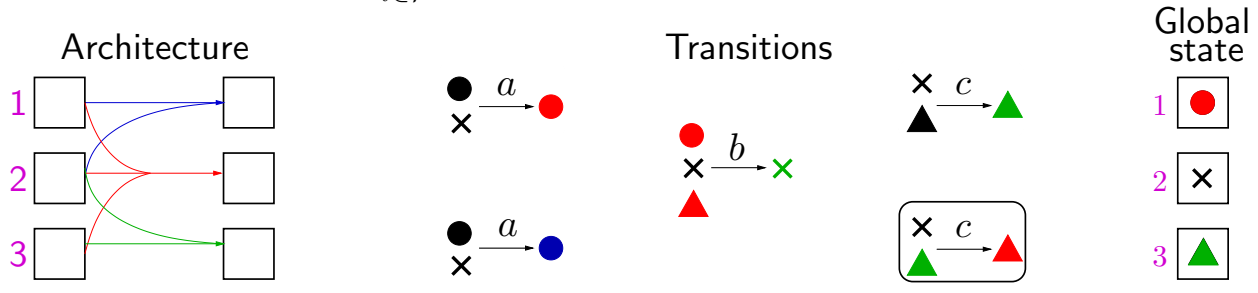
- $\forall i \in \mathcal{P}$, Q_i is the set of local states for process i
- $\forall a \in \Sigma$, $T_a \subseteq Q_{R(a)} \times Q_{W(a)}$, possible local moves of action a
- $q^0 \in Q = \prod_{i \in \mathcal{P}} Q_i$ is the global starting position of G



Asynchronous distributed systems

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

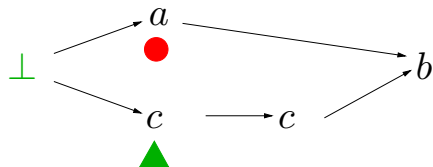
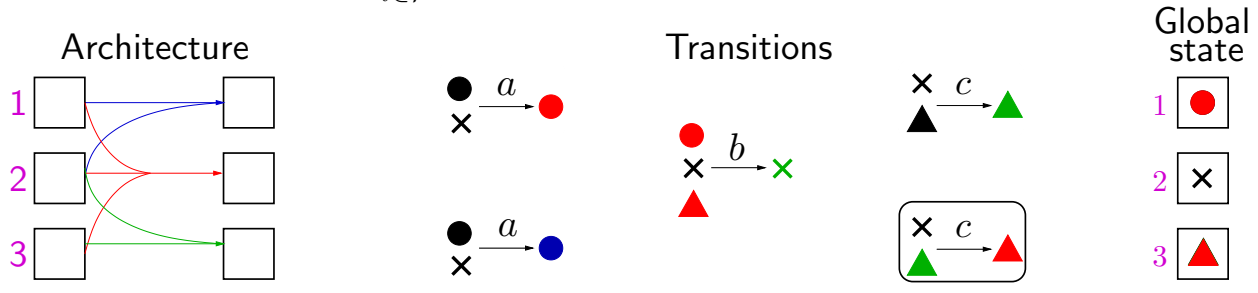
- $\forall i \in \mathcal{P}$, Q_i is the set of local states for process i
- $\forall a \in \Sigma$, $T_a \subseteq Q_{R(a)} \times Q_{W(a)}$, possible local moves of action a
- $q^0 \in Q = \prod_{i \in \mathcal{P}} Q_i$ is the global starting position of G



Asynchronous distributed systems

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

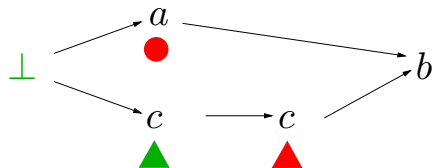
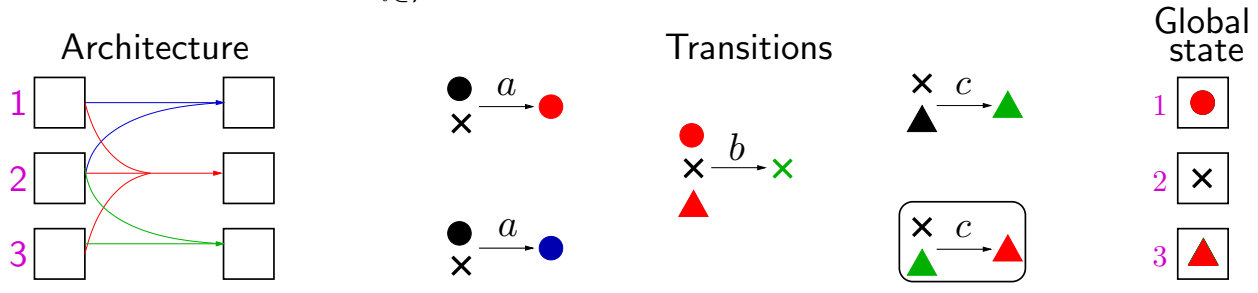
- $\forall i \in \mathcal{P}$, Q_i is the set of local states for process i
- $\forall a \in \Sigma$, $T_a \subseteq Q_{R(a)} \times Q_{W(a)}$, possible local moves of action a
- $q^0 \in Q = \prod_{i \in \mathcal{P}} Q_i$ is the global starting position of G



Asynchronous distributed systems

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

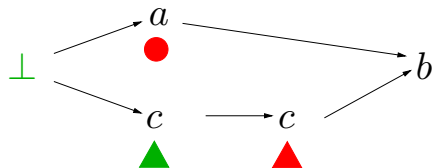
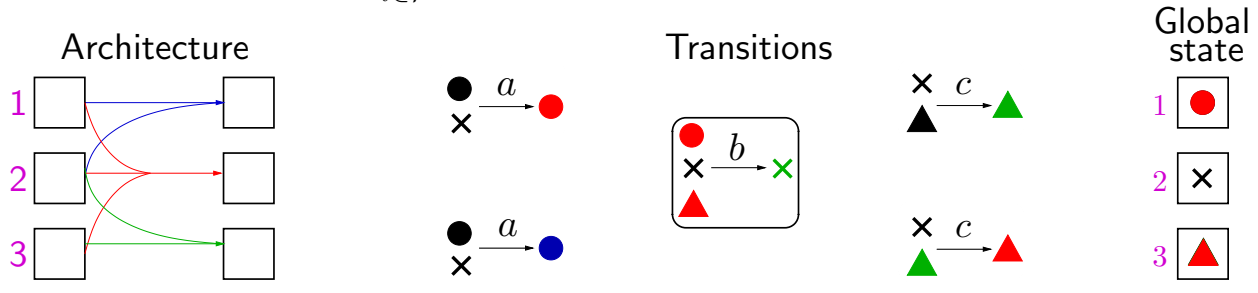
- $\forall i \in \mathcal{P}$, Q_i is the set of local states for process i
- $\forall a \in \Sigma$, $T_a \subseteq Q_{R(a)} \times Q_{W(a)}$, possible local moves of action a
- $q^0 \in Q = \prod_{i \in \mathcal{P}} Q_i$ is the global starting position of G



Asynchronous distributed systems

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

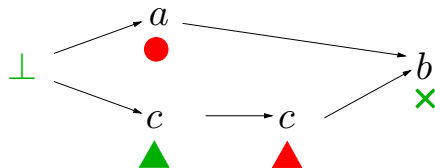
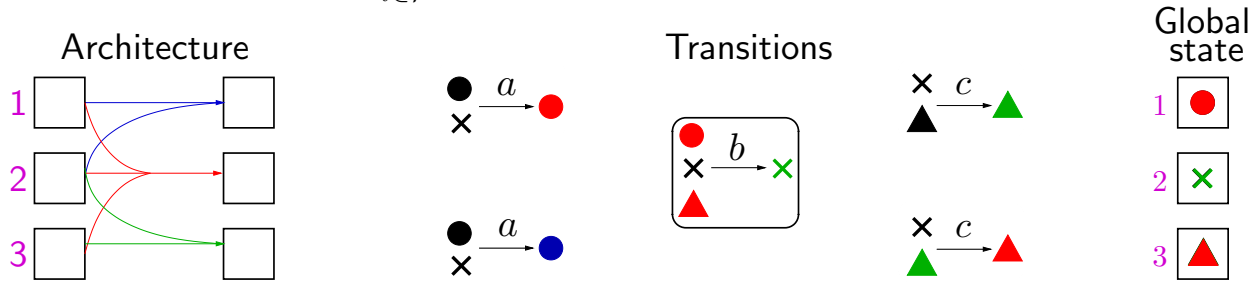
- $\forall i \in \mathcal{P}$, Q_i is the set of local states for process i
- $\forall a \in \Sigma$, $T_a \subseteq Q_{R(a)} \times Q_{W(a)}$, possible local moves of action a
- $q^0 \in Q = \prod_{i \in \mathcal{P}} Q_i$ is the global starting position of G



Asynchronous distributed systems

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

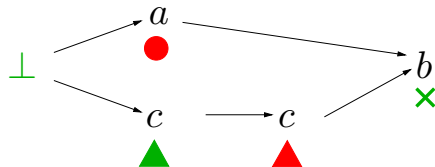
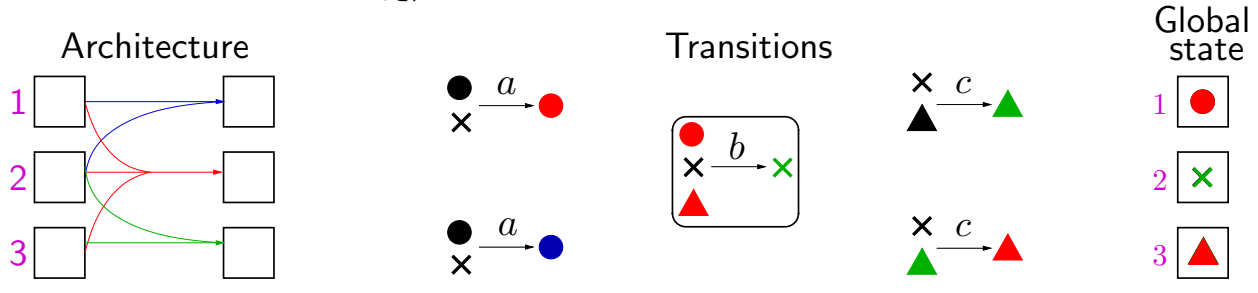
- $\forall i \in \mathcal{P}$, Q_i is the set of local states for process i
- $\forall a \in \Sigma$, $T_a \subseteq Q_{R(a)} \times Q_{W(a)}$, possible local moves of action a
- $q^0 \in Q = \prod_{i \in \mathcal{P}} Q_i$ is the global starting position of G



Asynchronous distributed systems

$\langle (Q_i)_{i \in \mathcal{P}}, (T_a)_{a \in \Sigma}, q^0 \rangle$ asynchronous distributed system 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 of G



Mazurkiewicz trace over $\Sigma' = \Sigma \times \prod Q_i$

Mazurkiewicz traces

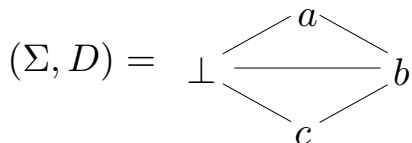
Dependence alphabet (Σ, D) with D reflexive & symmetric
Mazurkiewicz trace $t = (V, \leq, \lambda)$ where (V, \leq) poset and $\lambda : V \rightarrow \Sigma$ st.

$$\begin{cases} x \triangleleft y \Rightarrow \lambda(x) D \lambda(y) \\ \lambda(x) D \lambda(y) \Rightarrow x \leq y \text{ or } y \leq x \end{cases}$$

Mazurkiewicz traces

Dependence alphabet (Σ, D) with D reflexive & symmetric
Mazurkiewicz trace $t = (V, \leq, \lambda)$ where (V, \leq) poset and $\lambda : V \rightarrow \Sigma$ st.

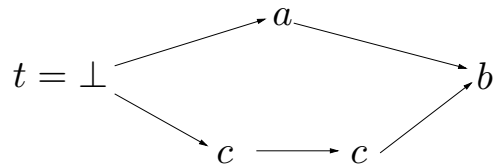
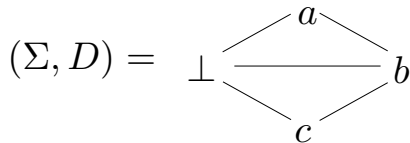
$$\begin{cases} x \leq y \Rightarrow \lambda(x) D \lambda(y) \\ \lambda(x) D \lambda(y) \Rightarrow x \leq y \text{ or } y \leq x \end{cases}$$



Mazurkiewicz traces

Dependence alphabet (Σ, D) with D reflexive & symmetric
Mazurkiewicz trace $t = (V, \leq, \lambda)$ where (V, \leq) poset and $\lambda : V \rightarrow \Sigma$ st.

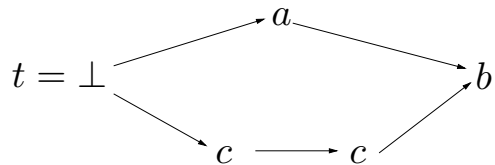
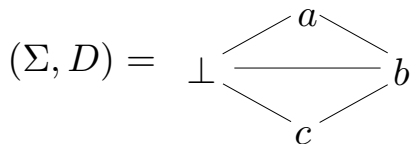
$$\begin{cases} x \leq y \Rightarrow \lambda(x) D \lambda(y) \\ \lambda(x) D \lambda(y) \Rightarrow x \leq y \text{ or } y \leq x \end{cases}$$



Mazurkiewicz traces

Dependence alphabet (Σ, D) with D reflexive & symmetric
 Mazurkiewicz trace $t = (V, \leq, \lambda)$ where (V, \leq) poset and $\lambda : V \rightarrow \Sigma$ st.

$$\begin{cases} x \leq y \Rightarrow \lambda(x) D \lambda(y) \\ \lambda(x) D \lambda(y) \Rightarrow x \leq y \text{ or } y \leq x \end{cases}$$



Technical restriction

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

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

Asynchronous distributed games — Semantics

- $\Sigma = \Sigma_0 \uplus \Sigma_1$: partition of the set of players (actions) in teams 0 and 1.
Players of team 0 cooperate together against team 1.

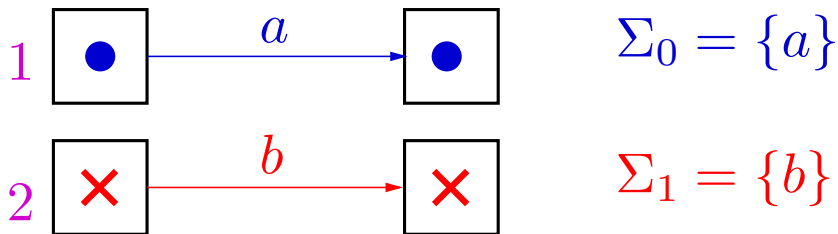
Asynchronous distributed games — Semantics

- $\Sigma = \Sigma_0 \uplus \Sigma_1$: partition of the set of players (actions) in teams 0 and 1.
Players of team 0 cooperate together against team 1.
- $\Sigma' = \{(a, p) \mid a \in \Sigma, p \in Q_{W(a)}\} \cup \{(\perp, q^0)\}$, $(a, p) D (b, q) \Leftrightarrow a D b$
- Positions of the game $V \subseteq \mathbb{M}(\Sigma', D)$.
- **Rules** of the game for player a given by T_a .
- A move $t \xrightarrow{a} (t \cdot (a, \vec{q}))$ corresponds to a progression in the rooted trace

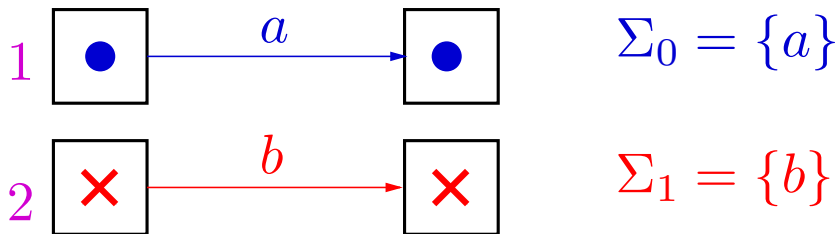
Asynchronous distributed games — Semantics

- $\Sigma = \Sigma_0 \uplus \Sigma_1$: partition of the set of players (actions) in teams 0 and 1.
Players of team 0 cooperate together against team 1.
- $\Sigma' = \{(a, p) \mid a \in \Sigma, p \in Q_{W(a)}\} \cup \{(\perp, q^0)\}$, $(a, p) D (b, q) \Leftrightarrow a D b$
- Positions of the game $V \subseteq \mathbb{M}(\Sigma', D)$.
- **Rules** of the game for player a given by T_a .
- A move $t \xrightarrow{a} (t \cdot (a, \vec{q}))$ corresponds to a progression in the rooted trace
- $\mathcal{W} \subseteq \mathbb{R}(\Sigma', D)$ is the **winning condition** of G .
Team 0 wins the play t iff $t \in \mathcal{W}$
- **Remarks** Teams 0 and 1 might be simultaneously enabled at $v \in V$.
One can recover the global state from a distributed play.

Asynchronous distributed games: example

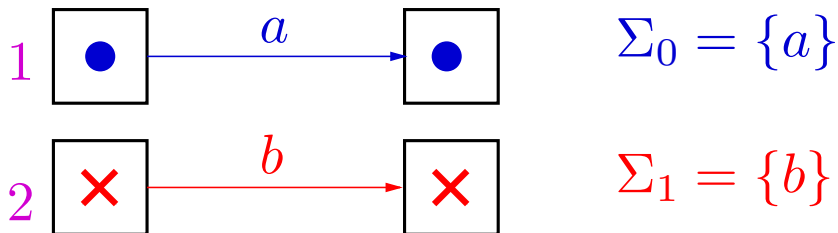


Asynchronous distributed games: example



$$\mathcal{W} = \mathbf{IM}(\Sigma, \mathbf{D}) \cup \{(\perp, q^0)(a, \bullet)^\omega (b, \times)^\omega\}$$

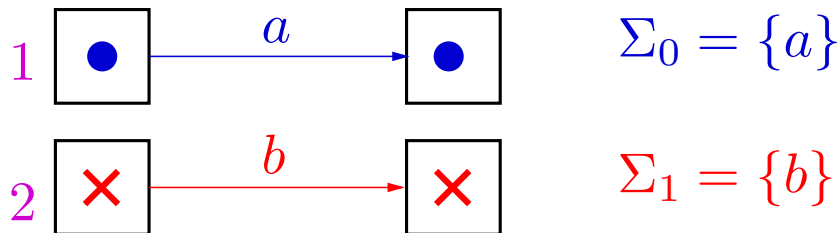
Asynchronous distributed games: example



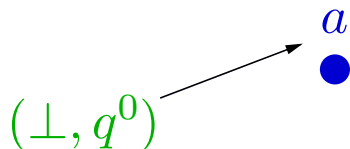
$$\mathcal{W} = \mathbb{IM}(\Sigma, \mathbf{D}) \cup \{(\perp, q^0)(a, \bullet)^\omega (b, \times)^\omega\}$$

$$(\perp, q^0)$$

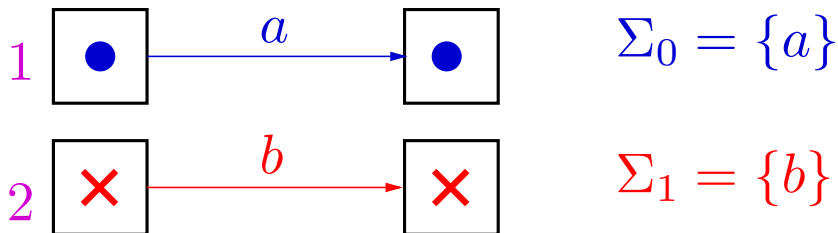
Asynchronous distributed games: example



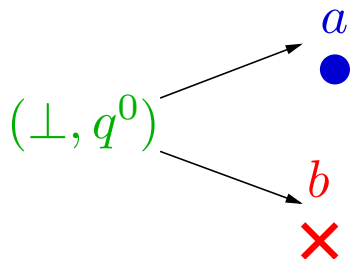
$$\mathcal{W} = \mathbb{IM}(\Sigma, \mathbf{D}) \cup \{(\perp, q^0)(a, \bullet)^\omega (b, \times)^\omega\}$$



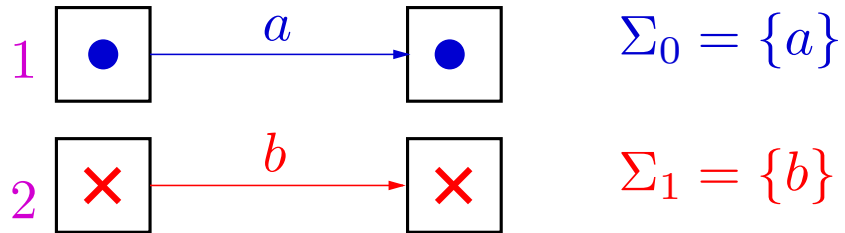
Asynchronous distributed games: example



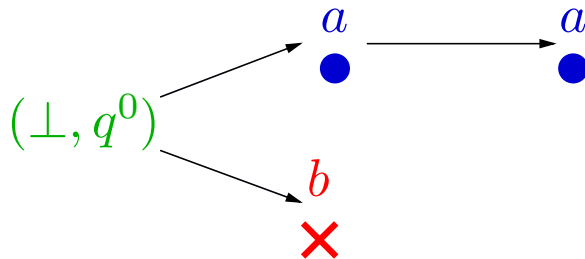
$$\mathcal{W} = \mathbb{IM}(\Sigma, \mathbf{D}) \cup \{(\perp, q^0)(a, \bullet)^\omega (b, \times)^\omega\}$$



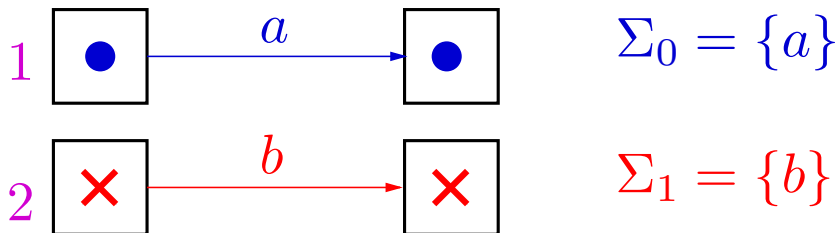
Asynchronous distributed games: example



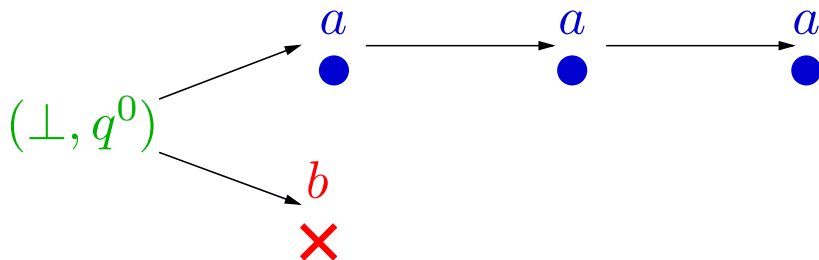
$$\mathcal{W} = \mathbb{M}(\Sigma, \mathbf{D}) \cup \{(\perp, q^0)(a, \bullet)^\omega (b, \times)^\omega\}$$



Asynchronous distributed games: example

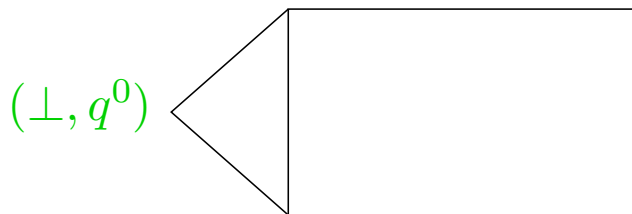


$$\mathcal{W} = \mathbb{IM}(\Sigma, \mathbf{D}) \cup \{(\perp, q^0)(a, \bullet)^\omega (b, \times)^\omega\}$$



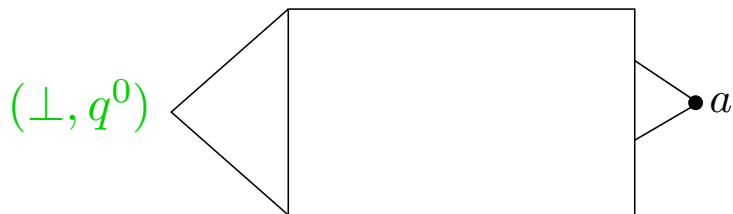
Distributed strategies with perfect memory

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



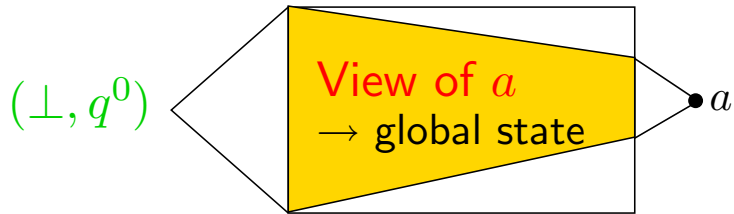
Distributed strategies with perfect memory

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



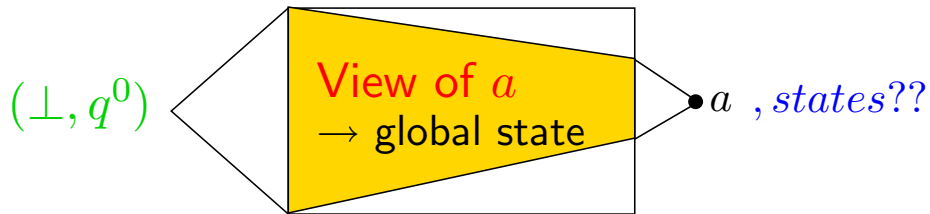
Distributed strategies with perfect memory

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



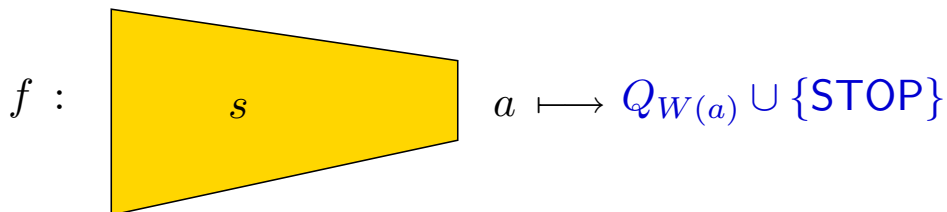
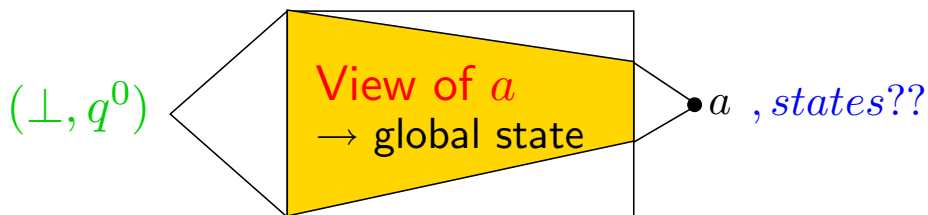
Distributed strategies with perfect memory

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



Distributed strategies with perfect memory

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



A distributed strategy f gives possible moves, hence must follow the rules.

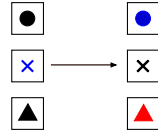
Computing strategies

- As in the classical setting, one can see
 - an ADG as a distributed program embedded in an environment.
 - ▷ Actions of Σ_0 can be controlled, actions of Σ_1 cannot.
 - a distributed play as an execution of the program.
 - the winning condition as a specification of good behaviors.
- Solving an asynchronous distributed game can be seen as computing a distributed controller for the asynchronous system.

Asynchronous distributed games & 2-players games

Global game vs. ADG

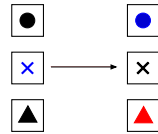
Global game = (global states, E)



$$\vec{p} \xrightarrow{a} \vec{q} \in E \text{ if } (\vec{x})_{R(a)}, \vec{y}_{W(a)} \in T_a \text{ and } \vec{q}_{\mathcal{P} \setminus W(a)} = \vec{p}_{\mathcal{P} \setminus W(a)}$$

Global game vs. ADG

Global game = (global states, E)



$$\vec{p} \xrightarrow{a} \vec{q} \in E \text{ if } (\vec{x})_{R(a)}, \vec{y}_{W(a)} \in T_a \text{ and } \vec{q}_{\mathcal{P} \setminus W(a)} = \vec{p}_{\mathcal{P} \setminus W(a)}$$

Facts

1. The global game may be determined while the ADG is not.
2. The distributed game has built-in perfect memory.
3. **Aim** From an ADG G , build a classical 2-players game H st.

$$\exists \text{ WDS for team 0 in } G \iff \exists \text{ WS for player 0 in } H$$

4. Global game not adequate

Back moves

The aim of player 1 is to show that team 0 does not have a WDS in G

- Addition of back moves for player 1 in the global game.
- \implies one has to build the memory **inside** the game H .

Positions of H : $S = \underbrace{\mathbb{M}(\Sigma', D')}_{\text{perfect memory}} \times \underbrace{(\Sigma_0 \cup \{1\})}_{\text{owner of the position}} + \dots$

On $(t, 1)$, the play so far is t and player 1 has to play.

On (t, a) , the play so far is t and player 0 has to make an a -move

Back moves

The aim of player 1 is to show that team 0 does not have a WDS in G

- Addition of back moves for player 1 in the global game.
- \implies one has to build the memory **inside** the game H .

Positions of H : $S = \underbrace{\mathbb{M}(\Sigma', D')}_{\text{perfect memory}} \times \underbrace{(\Sigma_0 \cup \{1\})}_{\text{owner of the position}} + \dots$

On $(t, 1)$, the play so far is t and player 1 has to play.

On (t, a) , the play so far is t and player 0 has to make an a -move

Initial position of the global game $s^0 = ((\perp, q^0), 1)$

Global play = $((\perp, q^0), 1) \rightarrow (t_0, a_0) \rightarrow (t_1, a_1) \rightarrow (t_2, a_2) \cdots$

From distributed games to sequential games

Moves of player 0

- $(t, a) \rightarrow (t \cdot (a, \vec{q}), 1)$ $a \in \Sigma_0$ and $t \xrightarrow{a} t \cdot (a, \vec{q})$ legal move in G .
- $(t, a) \rightarrow (t, 1)$. Player 0 refuses to make an a -move.

Moves of player 1

- $(t, 1) \rightarrow (t \cdot (a, q), 1)$ $a \in \Sigma_1$ and $t \xrightarrow{a} t \cdot (a, \vec{q})$ legal move in G .
- $(t, 1) \rightarrow (t, a)$ with $a \in \Sigma_0$. Player 1 asks player 0 to do an a -move.
- $(t \cdot (a, q), 1) \rightarrow (t, 1)$ with $a \neq \perp$. These **back-moves** are used to show that player 0 is not following a **distributed** strategy.

Winning condition in the sequential game

Winning condition in H : $\mathcal{W}+$ consistency + fairness

Consistency: “stay inside the same trace”

- with the same local view, player 0 has to play similarly.
- when player 1 asks player 0 to make an a -move, it should not force him to change for another trace.

Forbidden

- $(s^0, 1) \xrightarrow{1} (s^0, a) \xrightarrow{0} (s, 1) \xrightarrow{1} (s^0, 1) \xrightarrow{1} (s^0, b)$ where $a \text{ D } b$
- $(s^0, 1) \xrightarrow{1} (s^0, a) \xrightarrow{0} (s, 1) \xrightarrow{1} (s^0, 1) \xrightarrow{1} (s^0, a)$

Winning condition in the sequential game

Winning condition in H : $\mathcal{W}+$ consistency + fairness

Consistency: “stay inside the same trace”

- with the same local view, player 0 has to play similarly.
- when player 1 asks player 0 to make an a -move, it should not force him to change for another trace.

Forbidden

$$- (s^0, 1) \xrightarrow{1} (s^0, a) \xrightarrow{0} (s, 1) \xrightarrow{1} (s^0, 1) \xrightarrow{1} (s^0, b) \text{ where } a \text{ D } b$$

$$- (s^0, 1) \xrightarrow{1} (s^0, a) \xrightarrow{0} (s, 1) \xrightarrow{1} (s^0, 1) \xrightarrow{1} (s^0, a) \xrightarrow{0} (s', 1), s \neq s'$$

Winning condition in the sequential game

Winning condition in H : \mathcal{W} + consistency + fairness

Consistency: “stay inside the same trace”

- with the same local view, player 0 has to play similarly.
- when player 1 asks player 0 to make an a -move, it should not force him to change for another trace.

Forbidden

- $(s^0, 1) \xrightarrow{1} (s^0, a) \xrightarrow{0} (s, 1) \xrightarrow{1} (s^0, 1) \xrightarrow{1} (s^0, b)$ where $a \text{ D } b$
- $(s^0, 1) \xrightarrow{1} (s^0, a) \xrightarrow{0} (s, 1) \xrightarrow{1} (s^0, 1) \xrightarrow{1} (s^0, a) \xrightarrow{0} (s', 1)$, $s \neq s'$

Fairness condition: ensures that player 1 won't block the game
No more progress \Rightarrow player 0 was given the possibility (and refused)
to extend t with any possible move from Σ_0 .

From distributed games to sequential games

Thm There exists a WDS for team 0 in the distributed game G iff there exists a memoryless DS for player 0 in the global game H .

Problem Even if the ADG G is finite, H is infinite (built-in memory)

Open Does there exist, in some cases, a **finite** game

- trace alphabet (architecture)
- winning condition
- relationship between architecture and winning condition.

Asynchronous alternating automata

Asynchronous automata. Motivation

- **Asynchronous** automata used for specification.
- **Alternating** automata can also be used for specification.
Compilation of LTL formulas into alternating automata.
- Some branching properties of distributed systems cannot be captured when working on linearizations. One should be able to distinguish between
 - non-determinism
 - concurrency.

Alternating asynchronous automata (AAA)

Alternating automata: $\delta_a : Q_{R(a)} \rightarrow 2^{Q_{W(a)}}$

Alternating asynchronous automata: $\delta_a : Q_{R(a)} \rightarrow \mathcal{B}^+(Q_{W(a)})$

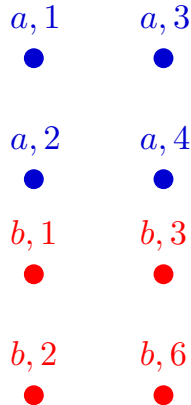
Runs: (prime) **well-formed** event structures.

$$R(a) = W(a) = \{1\} \quad \delta_a(0) = 1 \wedge 2 \quad \delta_b(0) = 1 \wedge 2$$

$$R(b) = W(b) = \{2\} \quad \delta_a(1) = 3 \vee 5 \quad \delta_b(1) = 3 \vee 5$$

$$\delta_a(2) = 4 \vee 6 \quad \delta_b(2) = 4 \vee 6$$

Run on *aabb*



Alternating asynchronous automata (AAA)

Alternating automata: $\delta_a : Q_{R(a)} \rightarrow 2^{Q_{W(a)}}$

Alternating asynchronous automata: $\delta_a : Q_{R(a)} \rightarrow \mathcal{B}^+(Q_{W(a)})$

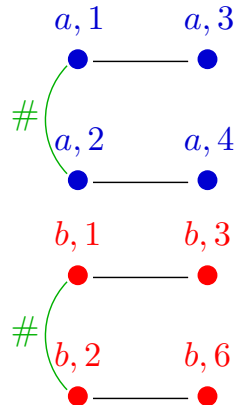
Runs: (prime) **well-formed** event structures.

$$R(a) = W(a) = \{1\} \quad \delta_a(0) = 1 \wedge 2 \quad \delta_b(0) = 1 \wedge 2$$

$$R(b) = W(b) = \{2\} \quad \delta_a(1) = 3 \vee 5 \quad \delta_b(1) = 3 \vee 5$$

$$\delta_a(2) = 4 \vee 6 \quad \delta_b(2) = 4 \vee 6$$

Run on *aabb*



Distributed game associated to an AAA

Prop Given an AAA \mathcal{A} and $t \in \mathbb{M}(\Sigma, D)$, \exists a **reachability** game $G(\mathcal{A}, t)$:
 $t \in \mathcal{L}(\mathcal{A})$ iff team 0 has a WDS in $G(\mathcal{A}, t)$

Distributed game associated to an AAA

Prop Given an AAA \mathcal{A} and $t \in \mathbb{M}(\Sigma, D)$, \exists a **reachability** game $G(\mathcal{A}, t)$:

$t \in \mathcal{L}(\mathcal{A})$ iff team 0 has a WDS in $G(\mathcal{A}, t)$

a,b independent	$\delta_a(0) = 1 \wedge 2$	$\delta_b(0) = 1 \wedge 2$
	$\delta_a(1) = 3 \vee 5$	$\delta_b(1) = 3 \vee 5$
	$\delta_a(2) = 4 \vee 6$	$\delta_b(2) = 4 \vee 6$

Distributed game associated to an AAA

Prop Given an AAA \mathcal{A} and $t \in \mathbb{M}(\Sigma, D)$, \exists a **reachability** game $G(\mathcal{A}, t)$:

$t \in \mathcal{L}(\mathcal{A})$ iff team 0 has a WDS in $G(\mathcal{A}, t)$

a,b independent

$$\delta_a(0) = 1 \wedge 2$$

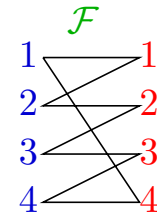
$$\delta_a(1) = 3 \vee 5$$

$$\delta_a(2) = 4 \vee 6$$

$$\delta_b(0) = 1 \wedge 2$$

$$\delta_b(1) = 3 \vee 5$$

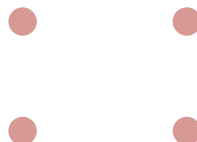
$$\delta_b(2) = 4 \vee 6$$



0



0



Distributed game associated to an AAA

Prop Given an AAA \mathcal{A} and $t \in \mathbb{M}(\Sigma, D)$, \exists a **reachability** game $G(\mathcal{A}, t)$:

$t \in \mathcal{L}(\mathcal{A})$ iff team 0 has a WDS in $G(\mathcal{A}, t)$

a,b independent

$$\delta_a(0) = 1 \wedge 2$$

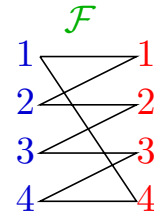
$$\delta_a(1) = 3 \vee 5$$

$$\delta_a(2) = 4 \vee 6$$

$$\delta_b(0) = 1 \wedge 2$$

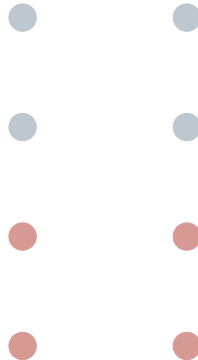
$$\delta_b(1) = 3 \vee 5$$

$$\delta_b(2) = 4 \vee 6$$



0

0



Distributed game associated to an AAA

Prop Given an AAA \mathcal{A} and $t \in \mathbb{M}(\Sigma, D)$, \exists a **reachability** game $G(\mathcal{A}, t)$:

$t \in \mathcal{L}(\mathcal{A})$ iff team 0 has a WDS in $G(\mathcal{A}, t)$

a,b independent

$$\delta_a(0) = 1 \wedge 2$$

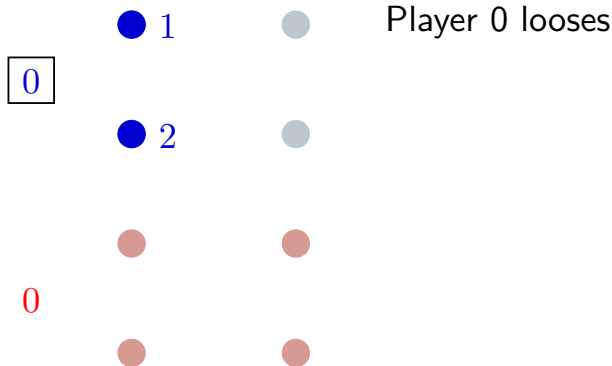
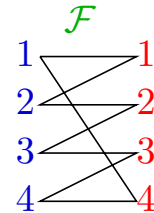
$$\delta_a(1) = 3 \vee 5$$

$$\delta_a(2) = 4 \vee 6$$

$$\delta_b(0) = 1 \wedge 2$$

$$\delta_b(1) = 3 \vee 5$$

$$\delta_b(2) = 4 \vee 6$$



Distributed game associated to an AAA

Prop Given an AAA \mathcal{A} and $t \in \mathbb{M}(\Sigma, D)$, \exists a **reachability** game $G(\mathcal{A}, t)$:

$t \in \mathcal{L}(\mathcal{A})$ iff team 0 has a WDS in $G(\mathcal{A}, t)$

a,b independent

$$\delta_a(0) = 1 \wedge 2$$

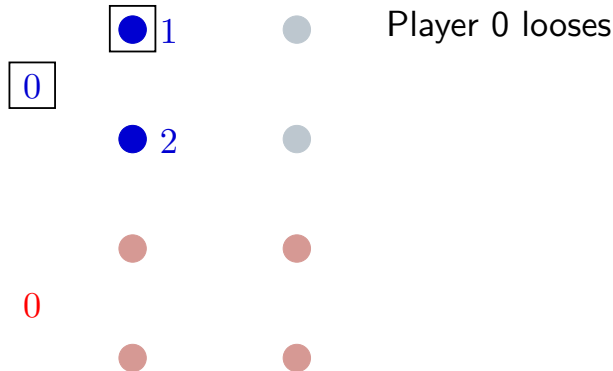
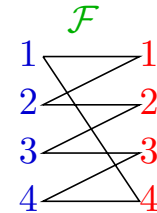
$$\delta_a(1) = 3 \vee 5$$

$$\delta_a(2) = 4 \vee 6$$

$$\delta_b(0) = 1 \wedge 2$$

$$\delta_b(1) = 3 \vee 5$$

$$\delta_b(2) = 4 \vee 6$$



Distributed game associated to an AAA

Prop Given an AAA \mathcal{A} and $t \in \mathbb{M}(\Sigma, D)$, \exists a **reachability** game $G(\mathcal{A}, t)$:

$t \in \mathcal{L}(\mathcal{A})$ iff team 0 has a WDS in $G(\mathcal{A}, t)$

a,b independent

$$\delta_a(0) = 1 \wedge 2$$

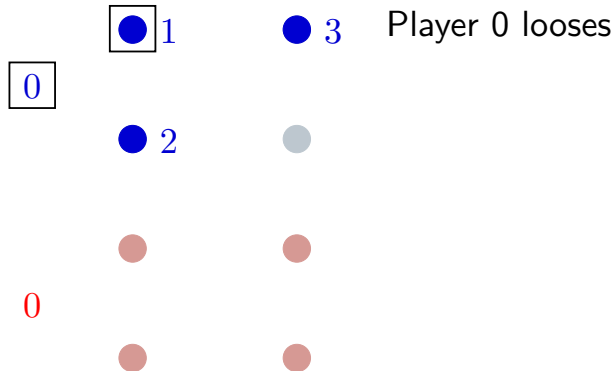
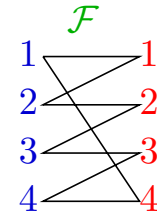
$$\delta_a(1) = 3 \vee 5$$

$$\delta_a(2) = 4 \vee 6$$

$$\delta_b(0) = 1 \wedge 2$$

$$\delta_b(1) = 3 \vee 5$$

$$\delta_b(2) = 4 \vee 6$$



Distributed game associated to an AAA

Prop Given an AAA \mathcal{A} and $t \in \mathbb{M}(\Sigma, D)$, \exists a **reachability** game $G(\mathcal{A}, t)$:

$t \in \mathcal{L}(\mathcal{A})$ iff team 0 has a WDS in $G(\mathcal{A}, t)$

a,b independent

$$\delta_a(0) = 1 \wedge 2$$

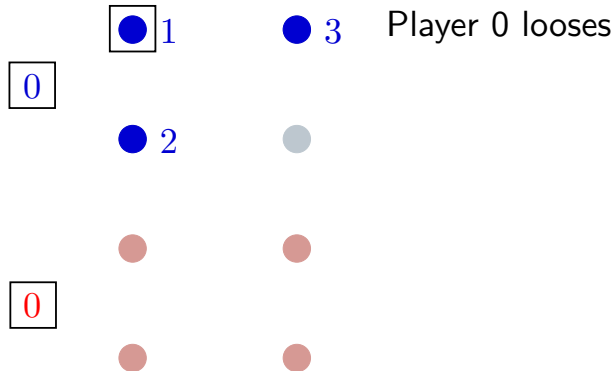
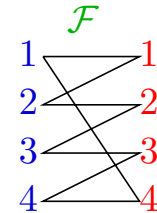
$$\delta_a(1) = 3 \vee 5$$

$$\delta_a(2) = 4 \vee 6$$

$$\delta_b(0) = 1 \wedge 2$$

$$\delta_b(1) = 3 \vee 5$$

$$\delta_b(2) = 4 \vee 6$$



Distributed game associated to an AAA

Prop Given an AAA \mathcal{A} and $t \in \mathbb{M}(\Sigma, D)$, \exists a **reachability** game $G(\mathcal{A}, t)$:

$t \in \mathcal{L}(\mathcal{A})$ iff team 0 has a WDS in $G(\mathcal{A}, t)$

a,b independent

$$\delta_a(0) = 1 \wedge 2$$

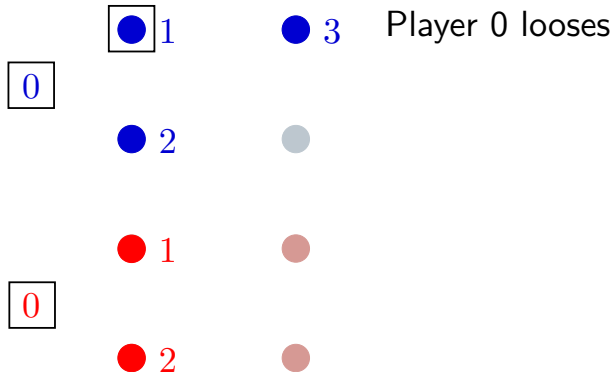
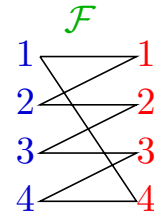
$$\delta_a(1) = 3 \vee 5$$

$$\delta_a(2) = 4 \vee 6$$

$$\delta_b(0) = 1 \wedge 2$$

$$\delta_b(1) = 3 \vee 5$$

$$\delta_b(2) = 4 \vee 6$$



Distributed game associated to an AAA

Prop Given an AAA \mathcal{A} and $t \in \mathbb{M}(\Sigma, D)$, \exists a **reachability** game $G(\mathcal{A}, t)$:

$t \in \mathcal{L}(\mathcal{A})$ iff team 0 has a WDS in $G(\mathcal{A}, t)$

a,b independent

$$\delta_a(0) = 1 \wedge 2$$

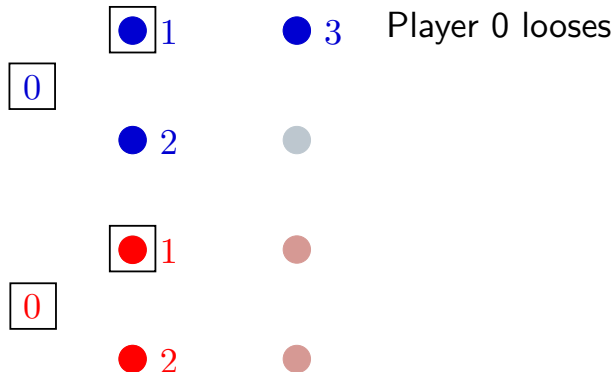
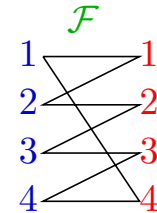
$$\delta_a(1) = 3 \vee 5$$

$$\delta_a(2) = 4 \vee 6$$

$$\delta_b(0) = 1 \wedge 2$$

$$\delta_b(1) = 3 \vee 5$$

$$\delta_b(2) = 4 \vee 6$$



Distributed game associated to an AAA

Prop Given an AAA \mathcal{A} and $t \in \mathbb{M}(\Sigma, D)$, \exists a **reachability** game $G(\mathcal{A}, t)$:

$t \in \mathcal{L}(\mathcal{A})$ iff team 0 has a WDS in $G(\mathcal{A}, t)$

a,b independent

$$\delta_a(0) = 1 \wedge 2$$

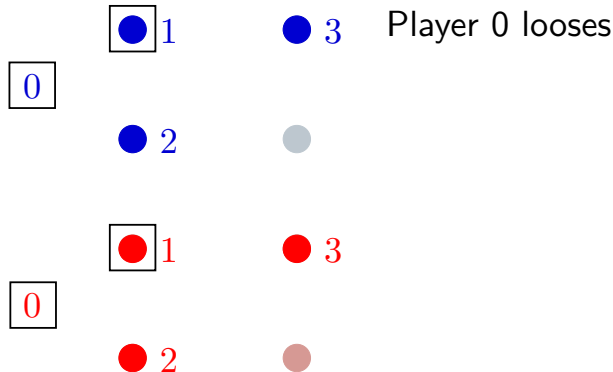
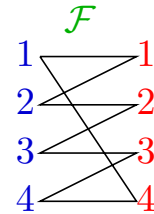
$$\delta_a(1) = 3 \vee 5$$

$$\delta_a(2) = 4 \vee 6$$

$$\delta_b(0) = 1 \wedge 2$$

$$\delta_b(1) = 3 \vee 5$$

$$\delta_b(2) = 4 \vee 6$$



Distributed game associated to an AAA

Prop Given an AAA \mathcal{A} and $t \in \mathbb{M}(\Sigma, D)$, \exists a **reachability** game $G(\mathcal{A}, t)$:

$t \in \mathcal{L}(\mathcal{A})$ iff team 0 has a WDS in $G(\mathcal{A}, t)$

a,b independent

$$\delta_a(0) = 1 \wedge 2$$

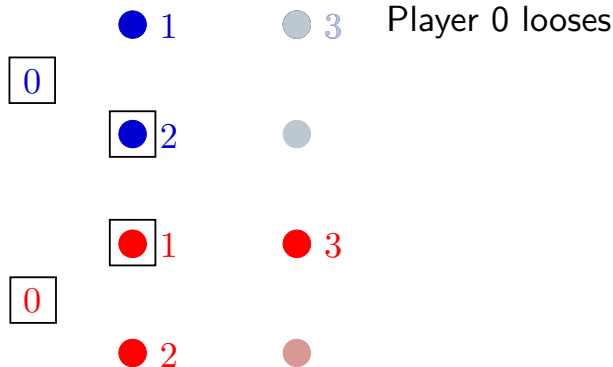
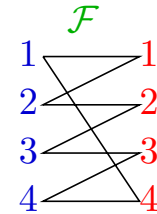
$$\delta_a(1) = 3 \vee 5$$

$$\delta_a(2) = 4 \vee 6$$

$$\delta_b(0) = 1 \wedge 2$$

$$\delta_b(1) = 3 \vee 5$$

$$\delta_b(2) = 4 \vee 6$$



Distributed game associated to an AAA

Prop Given an AAA \mathcal{A} and $t \in \mathbb{M}(\Sigma, D)$, \exists a **reachability** game $G(\mathcal{A}, t)$:

$t \in \mathcal{L}(\mathcal{A})$ iff team 0 has a WDS in $G(\mathcal{A}, t)$

a,b independent

$$\delta_a(0) = 1 \wedge 2$$

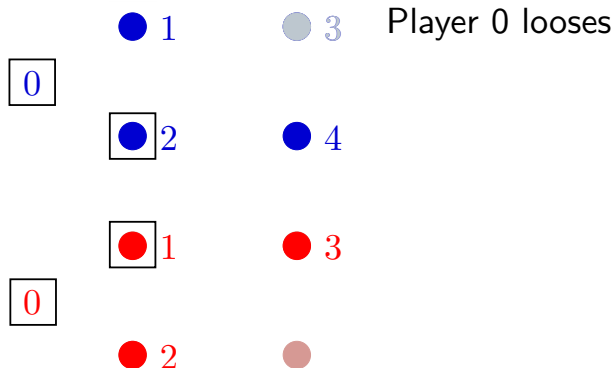
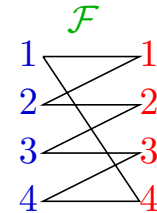
$$\delta_a(1) = 3 \vee 5$$

$$\delta_a(2) = 4 \vee 6$$

$$\delta_b(0) = 1 \wedge 2$$

$$\delta_b(1) = 3 \vee 5$$

$$\delta_b(2) = 4 \vee 6$$



Distributed game associated to an AAA

Prop Given an AAA \mathcal{A} and $t \in \mathbb{M}(\Sigma, D)$, \exists a **reachability** game $G(\mathcal{A}, t)$:

$t \in \mathcal{L}(\mathcal{A})$ iff team 0 has a WDS in $G(\mathcal{A}, t)$

a,b independent

$$\delta_a(0) = 1 \wedge 2$$

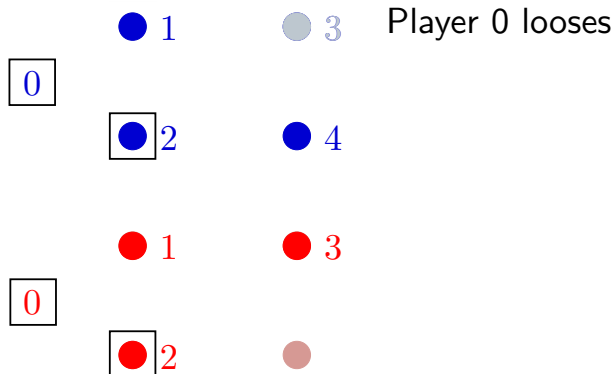
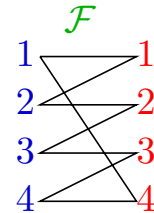
$$\delta_a(1) = 3 \vee 5$$

$$\delta_a(2) = 4 \vee 6$$

$$\delta_b(0) = 1 \wedge 2$$

$$\delta_b(1) = 3 \vee 5$$

$$\delta_b(2) = 4 \vee 6$$



Distributed game associated to an AAA

Prop Given an AAA \mathcal{A} and $t \in \mathbb{M}(\Sigma, D)$, \exists a **reachability** game $G(\mathcal{A}, t)$:

$t \in \mathcal{L}(\mathcal{A})$ iff team 0 has a WDS in $G(\mathcal{A}, t)$

a,b independent

$$\delta_a(0) = 1 \wedge 2$$

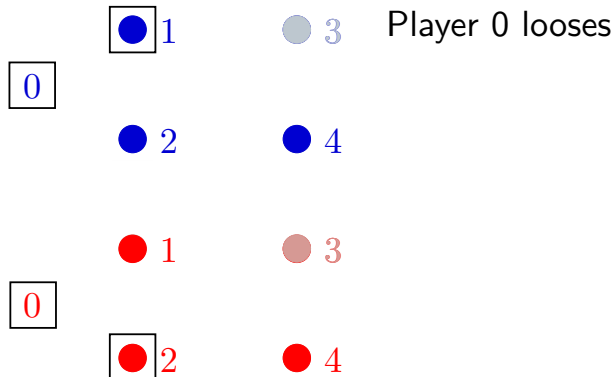
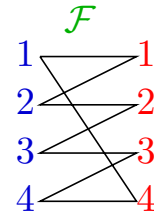
$$\delta_a(1) = 3 \vee 5$$

$$\delta_a(2) = 4 \vee 6$$

$$\delta_b(0) = 1 \wedge 2$$

$$\delta_b(1) = 3 \vee 5$$

$$\delta_b(2) = 4 \vee 6$$



Further work, open questions...

- In which cases can we find a finite quotient of the sequential game?
- Games are expected to be undecidable
Special trace languages \mathcal{W} where decidability holds?
- Relationship between other types of distributed games.
- Locate in the Borel hierarchy the set of fair plays and the set of consistent plays.
- Recognizability of the language accepted by an AAA?
- What about memoryless/bounded memory distributed strategies?
- Generalization of thm on AAA to infinite traces?

Related work

Related work

- [1] W. Zielonka. Notes on finite asynchronous automata. *RAIRO Inform. Théor. Appl.*, 21:99–135, 1987.
- [2] W. Zielonka. Asynchronous automata. In G. Rozenberg and V. Diekert, editors, *Book of Traces*, pages 175–217. World Scientific, 1995.
- [3] A. Pnueli and R. Rosner. Distributed reactive systems are hard to synthesize. In *Proc of the 31th FOCS*, pages 746–757, 1990.
- [4] M. Nielsen and C. Clausen. Games and logics for a noninterleaving bisimulation. *Nordic J. Comput.* **2**(2) (1995).
- [5] M. Mukund. Hereditary history preserving bisimulation is decidable for trace-labeled systems. *LNCS FSTTCS'02* (2002).
- [6] P. Madhusudan and P. S. Thiagarajan. Distributed controller synthesis for local specifications. *LNCS 2076*, 2001.
- [7] M. Jurdziński and M. Nielsen. Hereditary history preserving bisimilarity is undecidable. *Lect. Notes Comp. Sci. STACS'00* (2000).
- [8] M. Jurdziński. Deciding the winner in parity games is in $UP \cap co-UP$. *Information Processing Letters*, 68(3):119–124, November 1998.
- [9] M. Bednarczyk. Hereditary history preserving bisimulations. Tech. report, Polish Academy of Sciences, Gdansk (1991).