Concurrent Stochastic Lossy Channel Games

Daniel Stan

EPITA Research Lab[1], France

February 21, 2024

joint work with **Muhammad Najib**[2], **Anthony W. Lin**[3,4], **Parosh Abdulla**[5]

1 LRE LABORATOIRE DE RECHERCHE DE L'EPITA
2 HERIOT WATT UNIVERSITY
3 RPTU
4 MAX PLANCK INSTITUTE FOR SOFTWARE SYSTEMS
5 UPPSALA UNIVERSITET

Outline

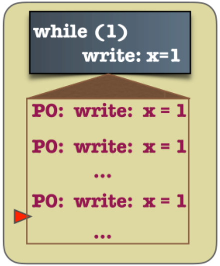
1. Lossy Channel Systems
2. Finite Concurrent Games
3. Infinite Concurrent Games

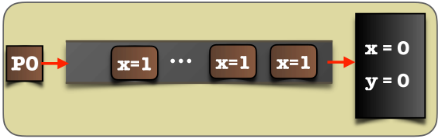
Channel Systems

Channel Systems (FIFO): Motivations

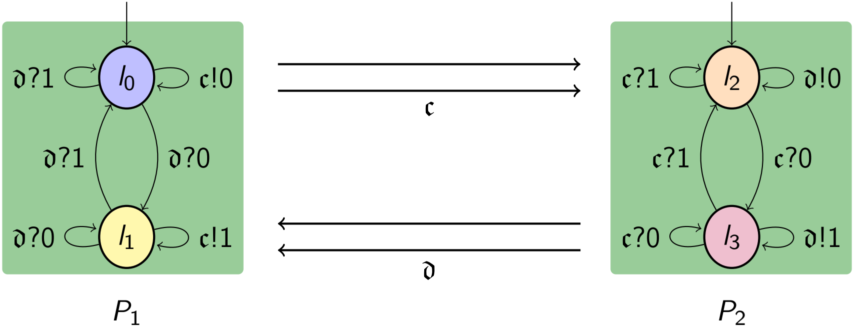
Modelisation and verification of systems with:

- Network transmissions;
- Transactional operations;
- TSO semantics [AABN18]

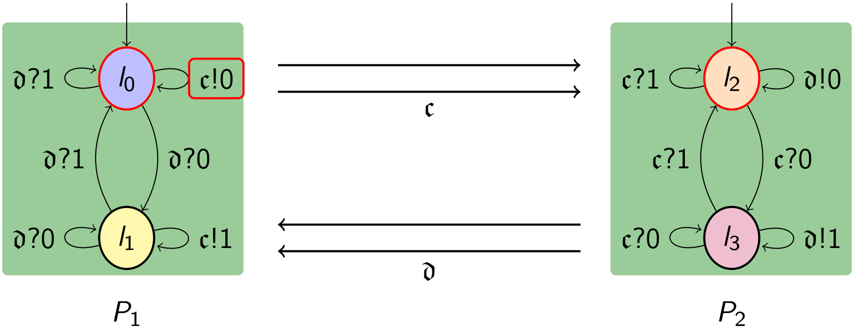




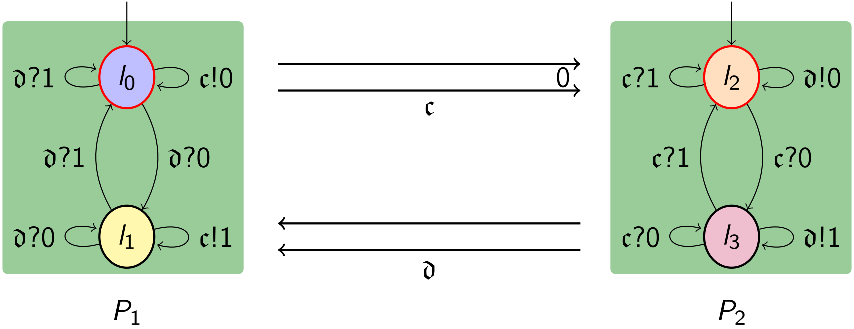
Channel Systems (FIFO): Motivations



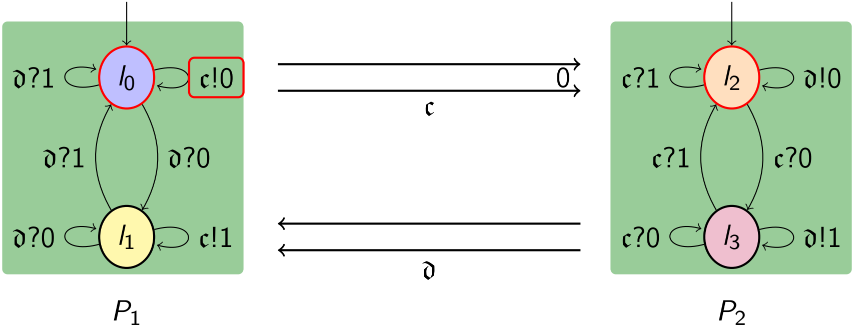
Channel Systems (FIFO): Motivations



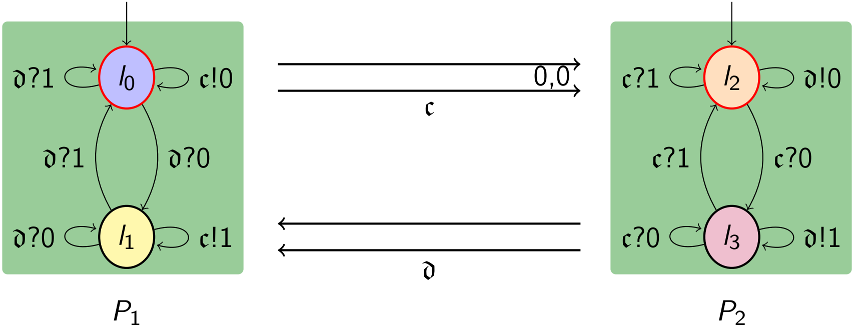
Channel Systems (FIFO): Motivations



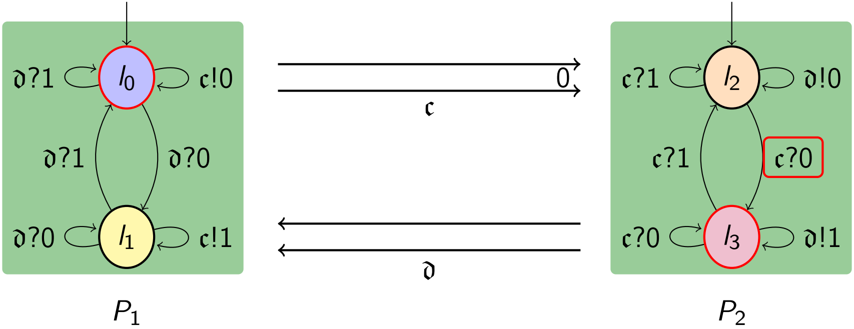
Channel Systems (FIFO): Motivations



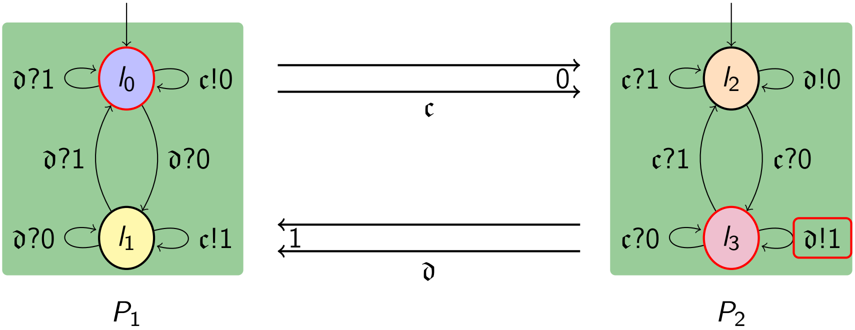
Channel Systems (FIFO): Motivations



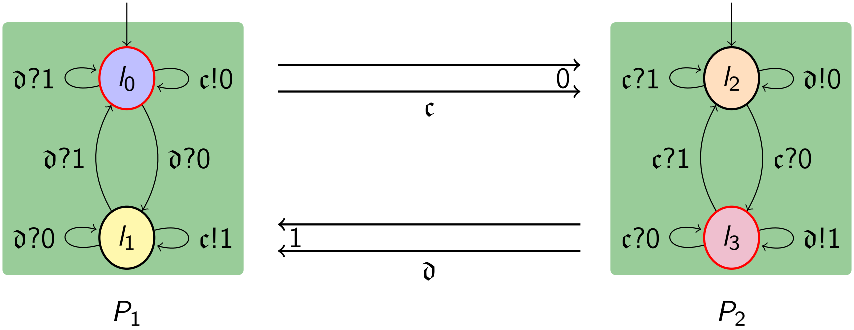
Channel Systems (FIFO): Motivations



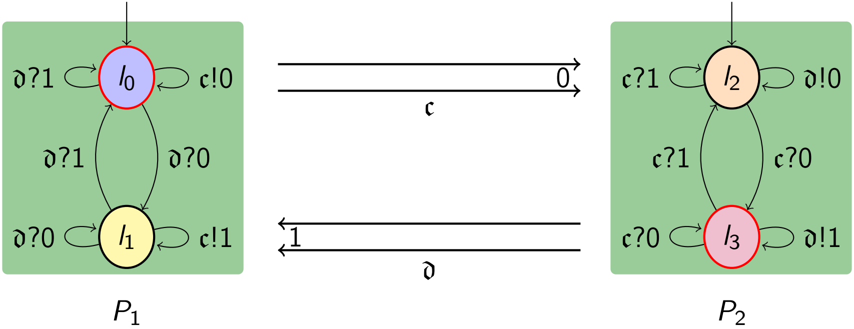
Channel Systems (FIFO): Motivations



Channel Systems (FIFO): Motivations



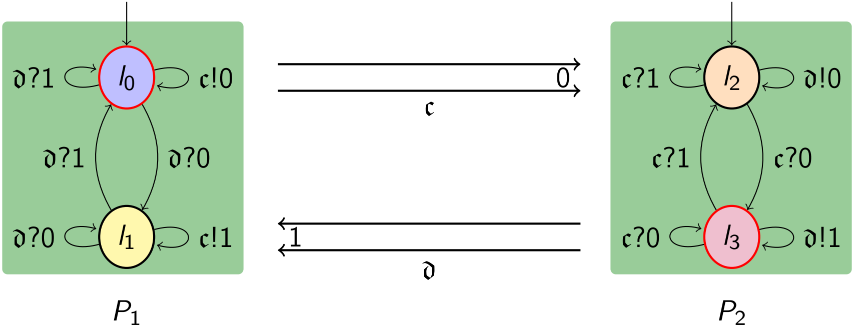
Channel Systems (FIFO): Motivations



Communication:

- Send a message m on $\mathfrak{c}$: $\mathfrak{c}!m$
- Receive a message $\mathfrak{c}?m$, **only if** m **was at the end of the queue.**

Channel Systems (FIFO): Motivations



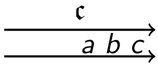
Communication:

- **Send** a message m on $\mathfrak{c}$: $\mathfrak{c}!m$
- **Receive** a message $\mathfrak{c}?m$, **only if** m **was at the end of the queue**.

For this talk: only **one** channel, and **one** component.

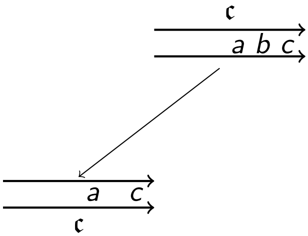
Lossiness assumption

Assumption: at every round, every message *may* disappear.



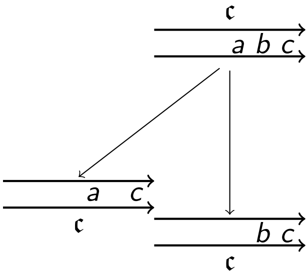
Lossiness assumption

Assumption: at every round, every message *may* disappear.



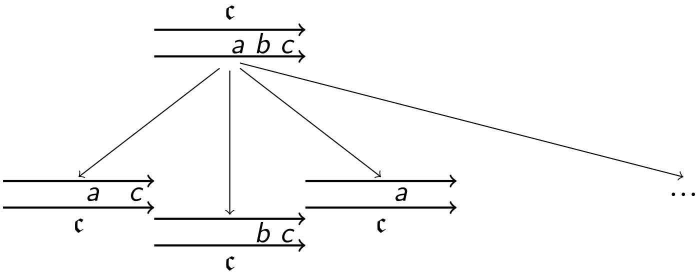
Lossiness assumption

Assumption: at every round, every message *may* disappear.



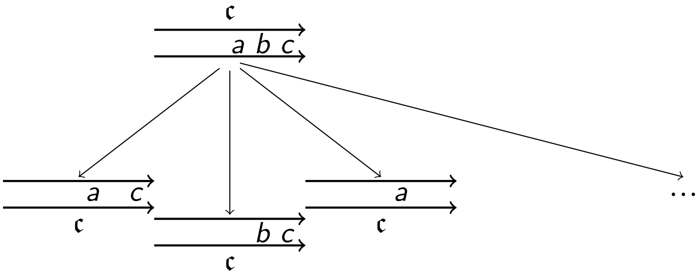
Lossiness assumption

Assumption: at every round, every message *may* disappear.



Lossiness assumption

Assumption: at every round, every message *may* disappear.



Effect of a transition $l \xrightarrow{f} l'$ on state $s = l \cdot w$:

1. Change location to l' ;
2. Apply the channel operation f on w to get w';
3. Drop from w' to $w'' \preceq w'$: subword ordering.

Result: $s' = l' \cdot w''$:

Subword-ordering $\preceq$ is a well-quasi order [FS01]

Definition ([FS01])

$(S, \preceq)$ is a well-quasi-order (WQO) if: $\forall (s_i)_{i \in \mathbb{N}} \in S^{\mathbb{N}}, \exists i < j : s_i \preceq s_j$

Subword-ordering $\preceq$ is a well-quasi order [FS01]

Definition ([FS01])

$(S, \preceq)$ is a well-quasi-order (WQO) if: $\forall (s_i)_{i \in \mathbb{N}} \in S^{\mathbb{N}}, \exists i < j : s_i \preceq s_j$
$(S, \rightarrow, \preceq)$ is a well-structured transition system:

$$\forall s, s', t, \quad \begin{array}{c} t \\ \curlyeqsucc \\ s \longrightarrow s' \end{array}$$

Subword-ordering $\preceq$ is a well-quasi order [FS01]

Definition ([FS01])

$(S, \preceq)$ is a well-quasi-order (WQO) if: $\forall (s_i)_{i \in \mathbb{N}} \in S^{\mathbb{N}}, \exists i < j : s_i \preceq s_j$

$(S, \to, \preceq)$ is a well-structured transition system:

$$\forall s, s', t, \quad \begin{array}{ccc} t & \longrightarrow & t'^{\,\exists} \\ \curlyvee\!\mid & & \curlyvee\!\mid \\ s & \longrightarrow & s' \end{array}$$

a.k.a. $\mathrm{Pre}(s) = \{t \mid s \to t\}$ preserves $\preceq$-closed sets

Subword-ordering $\preceq$ is a well-quasi order [FS01]

Definition ([FS01])

$(S, \preceq)$ is a well-quasi-order (WQO) if: $\forall (s_i)_{i \in \mathbb{N}} \in S^{\mathbb{N}}, \exists i < j : s_i \preceq s_j$

$(S, \rightarrow, \preceq)$ is a well-structured transition system:

$$\forall s, s', t, \quad \begin{array}{ccc} t & \longrightarrow & t'^{\,\exists} \\ \curlyvee\!\!| & & \curlyvee\!\!| \\ s & \longrightarrow & s' \end{array}$$

a.k.a. $\mathrm{Pre}(s) = \{t \mid s \rightarrow t\}$ preserves $\preceq$-closed sets

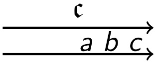
backward reachability scheme [FS01] for **non-deterministic schedulers**:

$$\bigcup_{n \geq 0} \mathrm{Pre}^n(R) = \{s \mid \exists (s_n) : s = s_0 \rightarrow s_1 \rightarrow \ldots s_k \in R\} = [\![\mathrm{E}(\Diamond R)]\!]$$

Lossiness in the probabilistic case

Stochastic case: the semantics is a Markov chain $(S, \Pr)$.

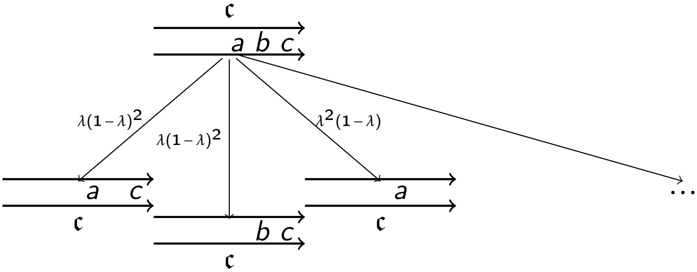
Local lossiness assumption: at every step, there is a positive probability $\lambda \in (0,1)$, that a letter in the channel is dropped. Every message drop event is **independent** from the others.



Lossiness in the probabilistic case

Stochastic case: the semantics is a Markov chain (S, Pr).

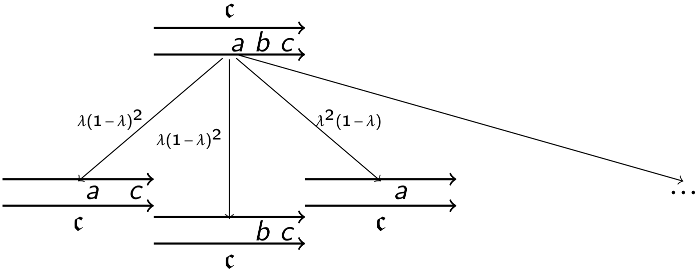
Local lossiness assumption: at every step, there is a positive probability $\lambda \in (0,1)$, that a letter in the channel is dropped. Every message drop event is **independent** from the others.



Lossiness in the probabilistic case

Stochastic case: the semantics is a Markov chain (S, Pr).

Local lossiness assumption: at every step, there is a positive probability $\lambda \in (0,1)$, that a letter in the channel is dropped. Every message drop event is **independent** from the others.

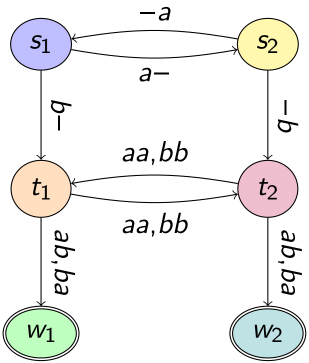


Qualitative setting:

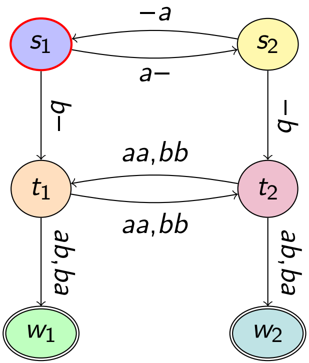
$[\![\mathrm{NZ}(\Diamond R)]\!] = \{s \mid \mathrm{Pr}(s \rightarrow^* s' \in R) > 0\}$ $[\![\mathrm{AS}(\Diamond R)]\!] = \{s \mid \mathrm{Pr}(s \rightarrow^* s' \in R) = 1\}$

Stochastic Concurrent Finite Games

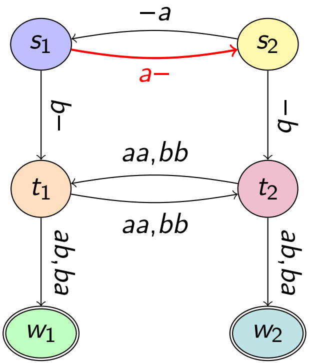
Concurrent Game on a Finite graph



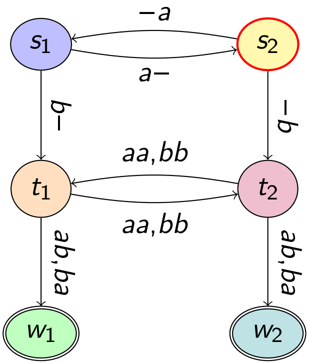
Concurrent Game on a Finite graph



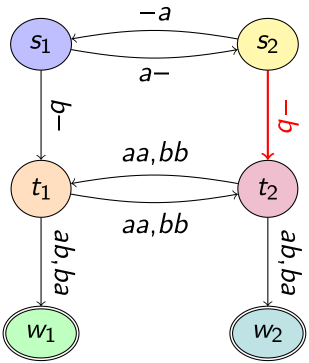
Concurrent Game on a Finite graph



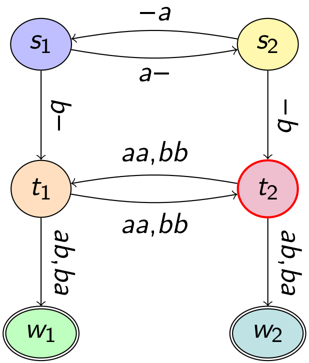
Concurrent Game on a Finite graph



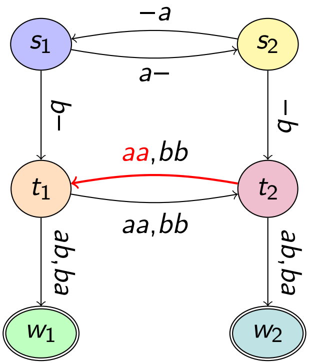
Concurrent Game on a Finite graph



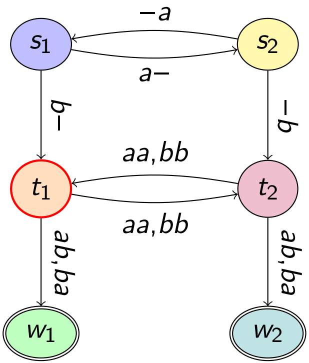
Concurrent Game on a Finite graph



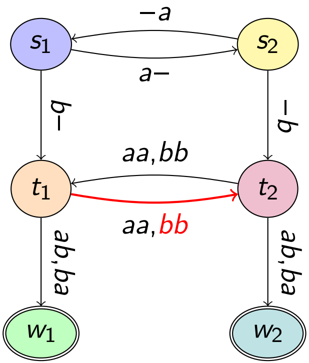
Concurrent Game on a Finite graph



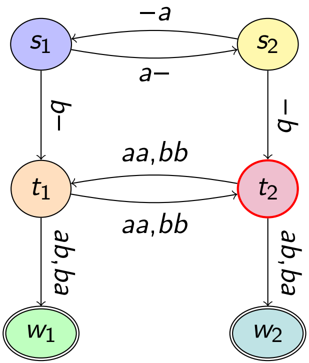
Concurrent Game on a Finite graph



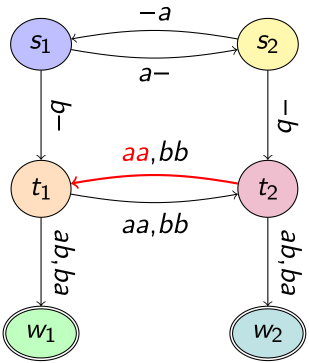
Concurrent Game on a Finite graph



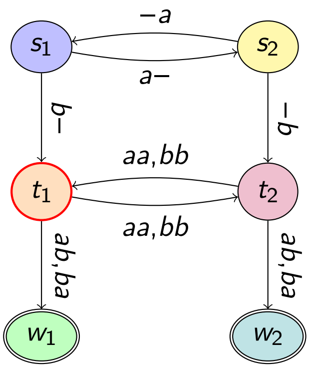
Concurrent Game on a Finite graph



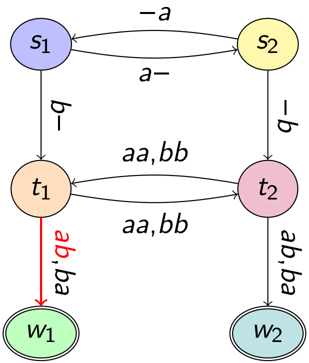
Concurrent Game on a Finite graph



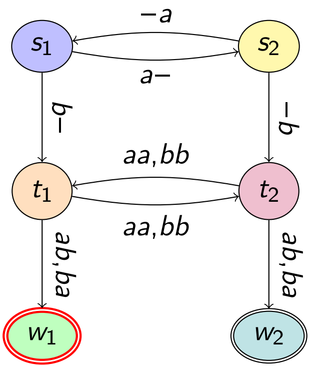
Concurrent Game on a Finite graph



Concurrent Game on a Finite graph

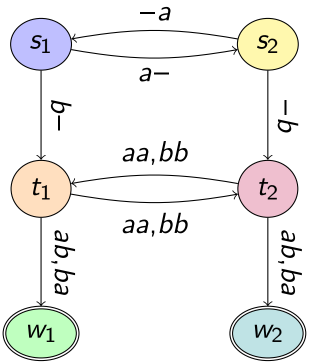


Concurrent Game on a Finite graph



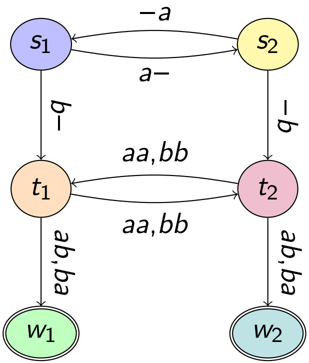
- Finite Game **Graph** Played by **multiple agents**
- Actions are played **concurrently**

Concurrent Game on a Finite graph



- Finite Game **Graph** Played by **multiple agents**
- Actions are played **concurrently**
- Also: **stochastic** transitions (players <u>and</u> environment)

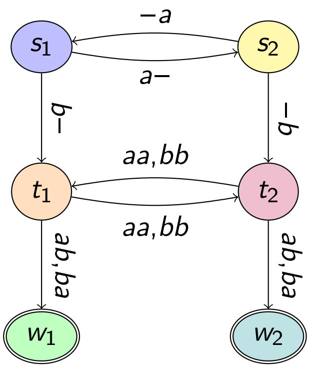
Concurrent Game on a Finite graph



- Finite Game **Graph** Played by **multiple agents**
- Actions are played **concurrently**
- Also: **stochastic** transitions (players and environment)
- Simple Objectives: **Reachability**, **Safety**, **Büchi**, **CoBüchi**:

  Ex: $\Diamond w_1$, $\Box\{w_2\}$, $\Box\Diamond s_1$, $\Diamond\Box\{t_1, t_2\}$

Concurrent Game on a Finite graph



- Finite Game **Graph** Played by **multiple agents**
- Actions are played **concurrently**
- Also: **stochastic** transitions (players and environment)
- Simple Objectives:
  Reachability, **Safety**, **Büchi**, **CoBüchi**:
  Ex: $\Diamond\, w_1$, $\Box\{w_2\}$, $\Box\Diamond\, s_1$, $\Diamond\Box\{t_1, t_2\}$
- Evaluated Qualitatively:
  almost surely, $\Pr(\ldots) = 1$ (AS)
  or **with positive probability** $\Pr(\ldots) > 0$ (NZ).

Different Ways of Winning

Players play strategies:

$$\forall i \in \mathrm{Agt},\ \sigma_i : \underbrace{s_0 \quad s_1 \ \ldots \ s_n}_{\text{history } \in S^+} \mapsto \delta \in \mathrm{Dist}(\underbrace{\mathrm{Act}_i(\ s_n \ )}_{\text{allowed actions in last state}})$$

Different Ways of Winning

Players play strategies:

$$\forall i \in \text{Agt},\ \sigma_i : \underbrace{s_0\ s_1\ \ldots\ s_n}_{\text{history } \in S^+} \mapsto \delta \in \text{Dist}(\underbrace{\text{Act}_i(\ s_n\ )}_{\text{allowed actions in last state}})$$

- **Zero-sum** case for two players, we compute the **winning regions**:
  $[\![\text{NZ}(\varphi_1)]\!]_1 = \{s \mid \exists \sigma_1 : \forall \sigma_2, \Pr^{\sigma_1,\sigma_2}(\varphi_1) > 0\}$
  $[\![\text{AS}(\varphi_1)]\!]_1 = \{s \mid \exists \sigma_1 : \forall \sigma_2, \Pr^{\sigma_1,\sigma_2}(\varphi_1) = 1\}$

Different Ways of Winning

Players play strategies:

$$\forall i \in \text{Agt},\ \sigma_i : \underbrace{s_0\ s_1\ \dots\ s_n}_{\text{history } \in S^+} \mapsto \delta \in \text{Dist}(\underbrace{\text{Act}_i(\ s_n\ )}_{\text{allowed actions in last state}})$$

- **Zero-sum** case for two players, we compute the **winning regions**:
  $[\![\text{NZ}(\varphi_1)]\!]_1 = \{s \mid \exists \sigma_1 : \forall \sigma_2, \Pr^{\sigma_1,\sigma_2}(\varphi_1) > 0\}$
  $[\![\text{AS}(\varphi_1)]\!]_1 = \{s \mid \exists \sigma_1 : \forall \sigma_2, \Pr^{\sigma_1,\sigma_2}(\varphi_1) = 1\}$
- For n players with objectives $(\Phi_i)_{i \in \text{Agt}}$ and a specification Γ, we consider the **Rational Verification** problem:
  Does there exists $\vec{\sigma}$ in the core satisfying Γ?

Different Ways of Winning

Players play strategies:

$$\forall i \in \mathrm{Agt},\ \sigma_i : \underbrace{s_0 \quad s_1 \ \dots \ s_n}_{\text{history } \in S^+} \mapsto \delta \in \mathrm{Dist}(\underbrace{\mathrm{Act}_i(\ s_n\ )}_{\text{allowed actions in last state}})$$

- **Zero-sum** case for two players, we compute the **winning regions**:
  $[\![\mathrm{NZ}(\varphi_1)]\!]_1 = \{s \mid \exists \sigma_1 : \forall \sigma_2, \Pr^{\sigma_1,\sigma_2}(\varphi_1) > 0\}$
  $[\![\mathrm{AS}(\varphi_1)]\!]_1 = \{s \mid \exists \sigma_1 : \forall \sigma_2, \Pr^{\sigma_1,\sigma_2}(\varphi_1) = 1\}$
- For n players with objectives $(\Phi_i)_{i \in \mathrm{Agt}}$ and a specification Γ, we consider the **Rational Verification** problem:

  Does there exists $\vec{\sigma}$ in the core satisfying Γ?

  Do all $\vec{\sigma}$ in the core satisfy Γ?

Different Ways of Winning

Players play strategies:

$$\forall i \in \text{Agt},\ \sigma_i : \underbrace{s_0\ s_1\ \dots\ s_n}_{\text{history } \in S^+} \mapsto \delta \in \text{Dist}(\underbrace{\text{Act}_i(\ s_n\ )}_{\text{allowed actions in last state}})$$

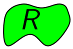
- **Zero-sum** case for two players, we compute the **winning regions**:
  $[\![\text{NZ}(\varphi_1)]\!]_1 = \{s \mid \exists \sigma_1 : \forall \sigma_2, \text{Pr}^{\sigma_1,\sigma_2}(\varphi_1) > 0\}$
  $[\![\text{AS}(\varphi_1)]\!]_1 = \{s \mid \exists \sigma_1 : \forall \sigma_2, \text{Pr}^{\sigma_1,\sigma_2}(\varphi_1) = 1\}$
- For n players with objectives $(\Phi_i)_{i \in \text{Agt}}$ and a specification Γ, we consider the **Rational Verification** problem:
  E-CORE: Does there exists $\vec{\sigma}$ in the core satisfying Γ?
  A-CORE: Do all $\vec{\sigma}$ in the core satisfy Γ?

Fixed Point Algorithms for Concurrent Games [dAHK07]

Assume: Two players, **Zero-sum**, Reachability Objective for a given set $R \subseteq S$ of states.
How to compute the **winning set** for player 1?

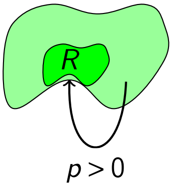
Recipe for $[\![\mathrm{NZ}(\Diamond R)]\!]_1$

Take $X := R$



Fixed Point Algorithms for Concurrent Games [dAHK07]

Assume: Two players, **Zero-sum**, Reachability Objective for a given set $R \subseteq S$ of states.
How to compute the **winning set** for player 1?

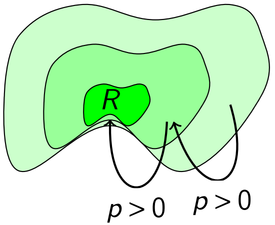


Recipe for $[\![\mathrm{NZ}(\Diamond R)]\!]_1$
Take $X := R$

// Add to X any state that 1 **can enforce**
// reaching **with positive probability**
$X := X \cup \mathrm{Pre}_1(X)$

Fixed Point Algorithms for Concurrent Games [dAHK07]

Assume: Two players, **Zero-sum**, Reachability Objective for a given set $R \subseteq S$ of states.
How to compute the **winning set** for player 1?



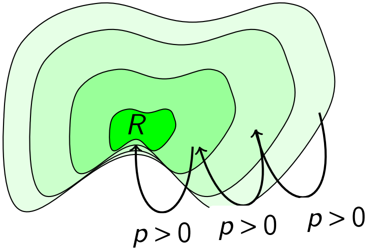
Recipe for $[\![\mathrm{NZ}(\Diamond R)]\!]_1$

Take $X := R$

// Add to X any state that 1 **can enforce**
// reaching **with positive probability**
$X := X \cup \mathrm{Pre}_1(X)$

Fixed Point Algorithms for Concurrent Games [dAHK07]

Assume: Two players, **Zero-sum**, Reachability Objective for a given set $R \subseteq S$ of states.
How to compute the **winning set** for player 1?



Recipe for $[\![\mathrm{NZ}(\Diamond R)]\!]_1$

Take $X := R$

Repeat until convergence:

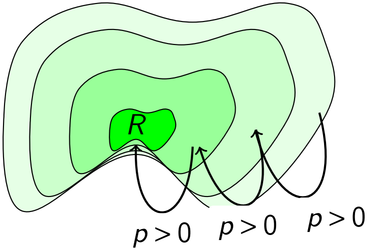
// Add to X any state that 1 **can enforce**

// reaching **with positive probability**

$X := X \cup \mathrm{Pre}_1(X)$

Fixed Point Algorithms for Concurrent Games [dAHK07]

Assume: Two players, **Zero-sum**, Reachability Objective for a given set $R \subseteq S$ of states.
How to compute the **winning set** for player 1?



Recipe for $[\![\mathrm{NZ}(\Diamond R)]\!]_1$

Take $X := R$

Repeat until convergence:

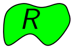
// Add to X any state that 1 **can enforce**
// reaching **with positive probability**
$X := X \cup \mathrm{Pre}_1(X)$

$\rightsquigarrow$ By **determinacy**, we can compute $[\![\mathrm{AS}(\Box R)]\!]_2$.

Fixed Point Algorithms for Concurrent Games [dAHK07] (bis)

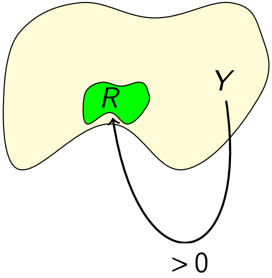
How about **Almost-Sure Reachability**?

Recipe for $[\![\mathrm{AS}(\Diamond R)]\!]_1$



Fixed Point Algorithms for Concurrent Games [dAHK07] (bis)

How about **Almost-Sure Reachability**?



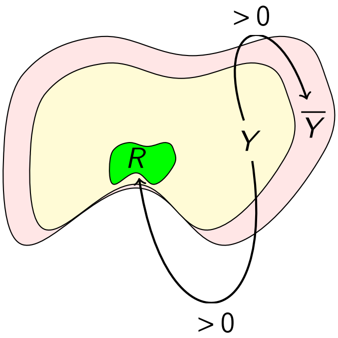
Recipe for $[\![\mathrm{AS}(\Diamond R)]\!]_1$

// Where can we force **positive probability**?

$Y := [\![\mathrm{NZ}(\Diamond R)]\!]_1$

Fixed Point Algorithms for Concurrent Games [dAHK07] (bis)

How about **Almost-Sure Reachability**?



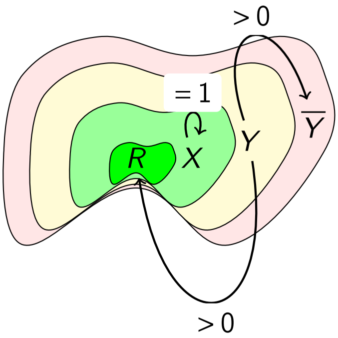
Recipe for $[\![\mathrm{AS}(\Diamond R)]\!]_1$

// Where can we force **positive probability**?

$Y := [\![\mathrm{NZ}(\Diamond R)]\!]_1$

Fixed Point Algorithms for Concurrent Games [dAHK07] (bis)

How about **Almost-Sure Reachability**?



Recipe for $[\![\mathrm{AS}(\Diamond R)]\!]_1$

// Where can we force **positive probability**?

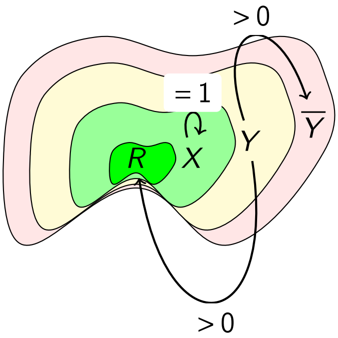
$Y := [\![\mathrm{NZ}(\Diamond R)]\!]_1$

// Now, **stay safe** in this set

$X := [\![\mathrm{AS}(\Box Y)]\!]_1$

Fixed Point Algorithms for Concurrent Games [dAHK07] (bis)

How about **Almost-Sure Reachability**?



Recipe for $[\![\mathrm{AS}(\Diamond R)]\!]_1$

// Where can we force **positive probability**?

$Y := [\![\mathrm{NZ}(\Diamond R)]\!]_1$

// Now, **stay safe** in this set

$X := [\![\mathrm{AS}(\Box Y)]\!]_1$

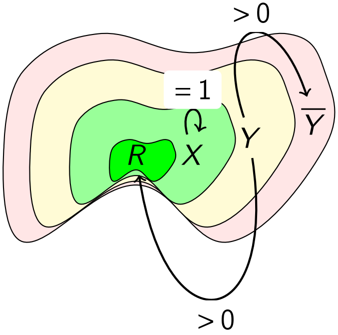
// ⚠ Remove actions losing for $\mathrm{AS}(\Box Y)$ ⚠

$\forall s\ \mathrm{Act}_1(s) := \{\alpha \in \mathrm{Act}_1(s) \mid \exists \beta : p(s,(\alpha,\beta),Y) < 1\}$

Repeat until convergence

Fixed Point Algorithms for Concurrent Games [dAHK07] (bis)

How about **Almost-Sure Reachability**?



Recipe for $[\![\mathrm{AS}(\Diamond R)]\!]_1$

// Where can we force **positive probability**?

$Y := [\![\mathrm{NZ}(\Diamond R)]\!]_1$

// Now, **stay safe** in this set

$X := [\![\mathrm{AS}(\Box Y)]\!]_1$

// ⚠Remove actions losing for $\mathrm{AS}(\Box Y)$ ⚠

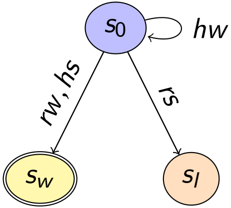
$\forall s\ \mathrm{Act}_1(s) := \{\alpha \in \mathrm{Act}_1(s) \mid \exists \beta : p(s,(\alpha,\beta),Y) < 1\}$

Repeat until convergence

$\rightsquigarrow$ By **determinacy**, we can compute $[\![\mathrm{NZ}(\Box R)]\!]_2$.

Example: Skirmish Game Analysis

- Step 1: $Y = X = \{\ s_0\ ,\ s_w\ \}$.

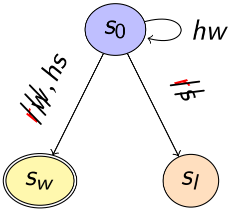


P1: $\mathrm{AS}(\Diamond\{\ s_w\ \})$

P2: $\mathrm{NZ}\left(\Box\overline{\{\ s_w\ \}}\right)$.

Example: Skirmish Game Analysis

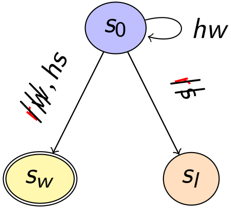
- Step 1: $Y = X = \{ s_0 , s_w \}$.
  But action r is losing. ⚠



P1: $\mathrm{AS}(\Diamond\{ s_w \})$

P2: $\mathrm{NZ}\left(\square\overline{\{ s_w \}}\right)$.

Example: Skirmish Game Analysis

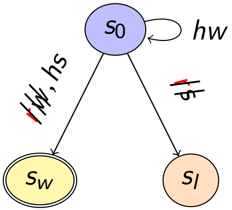


P1: $\mathrm{AS}(\Diamond\{s_w\})$

P2: $\mathrm{NZ}\left(\Box\overline{\{s_w\}}\right)$.

- Step 1: $Y = X = \{s_0, s_w\}$.
  But action r is losing. ⚠
- Step 2: $Y = X = \{s_w\} = [\![\mathrm{AS}(\Diamond s_w)]\!]_1$.

Example: Skirmish Game Analysis



P1: $\mathrm{AS}(\Diamond\{ s_w \})$

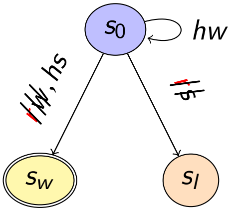
P2: $\mathrm{NZ}\left(\overline{\Box\{ s_w \}}\right)$.

- Step 1: $Y = X = \{ s_0 , s_w \}$.
  But action r is losing. ⚠
- Step 2: $Y = X = \{ s_w \} = [\![\mathrm{AS}(\Diamond\, s_w)]\!]_1$.

Some remarks:

- $\forall \epsilon > 0$; Player 1 can "win" with probability $1 - \epsilon$,
- For any **finite memory** strategy σ_2, player 1 can go to s_w **almost-surely**.
- Still, **Player 2** wins this game but with an **infinite memory strategy**.

Example: Skirmish Game Analysis



P1: $\mathrm{AS}(\Diamond\{\ s_w\ \})$

P2: $\mathrm{NZ}\left(\overline{\Box\{\ s_w\ \}}\right)$.

- Step 1: $Y = X = \{\ s_0\ ,\ s_w\ \}$.
  But action r is losing. ⚠
- Step 2: $Y = X = \{\ s_w\ \} = [\![\mathrm{AS}(\Diamond\ s_w\ )]\!]_1$.

Some remarks:

- $\forall \epsilon > 0$; Player 1 can "win" with probability $1 - \epsilon$,
- For any **finite memory** strategy σ_2, player 1 can go to s_w **almost-surely**.
- Still, **Player 2** wins this game but with an **infinite memory strategy**.

$$\forall n, \sigma_2(\underbrace{s_0 \ldots s_0}_{n \text{ times}})[s] = \left(\frac{1}{2}\right)^{2^{-n}}$$

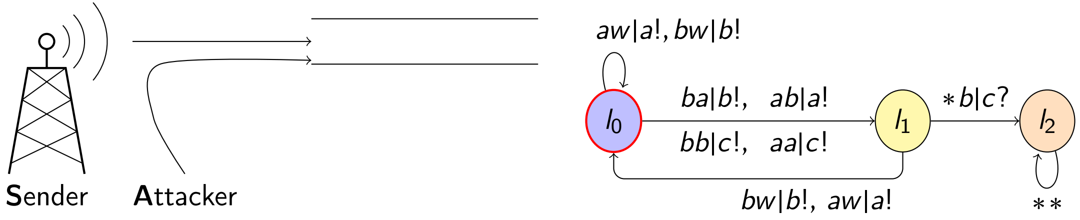
Concurrent Games + Lossy Channel Systems

Concurrent Games + Lossy Channel Systems

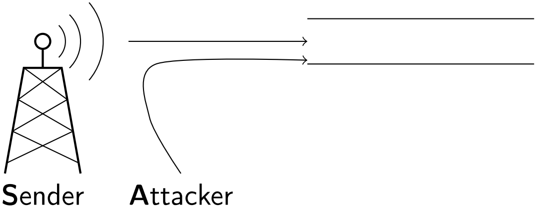
=

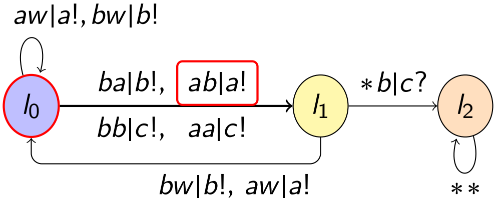
Infinite State Games

CSLCG: Def by Example

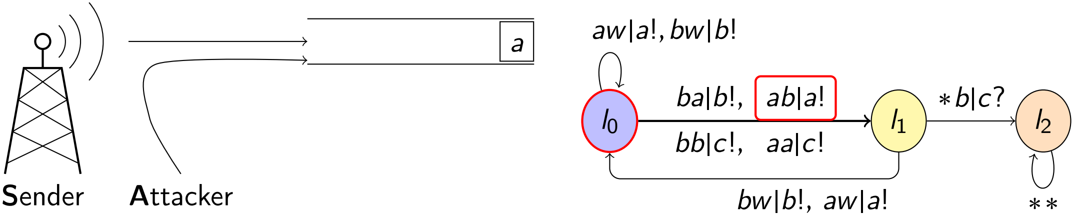


CSLCG: Def by Example

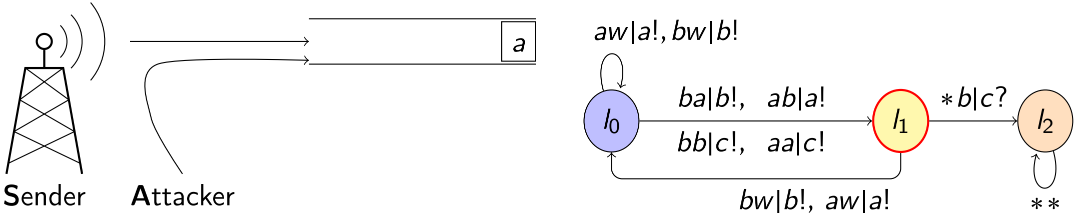




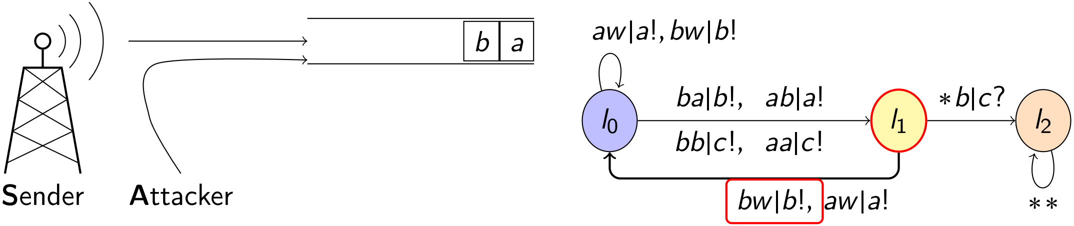
CSLCG: Def by Example



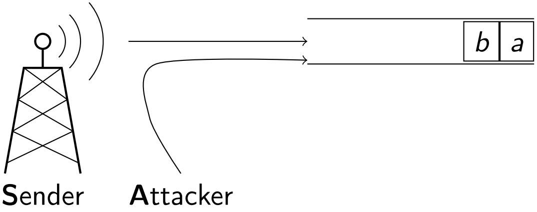
CSLCG: Def by Example

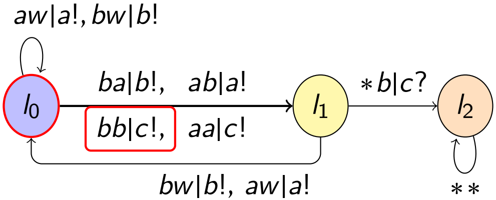


CSLCG: Def by Example

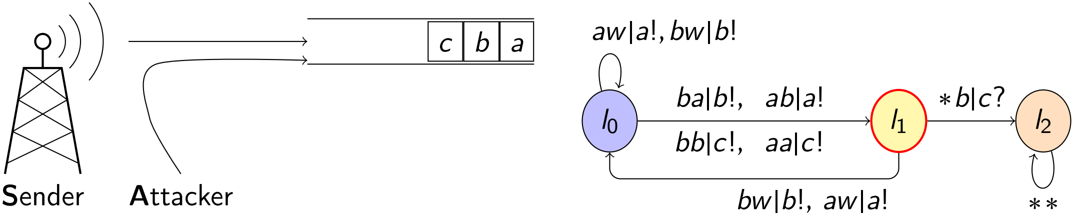


CSLCG: Def by Example

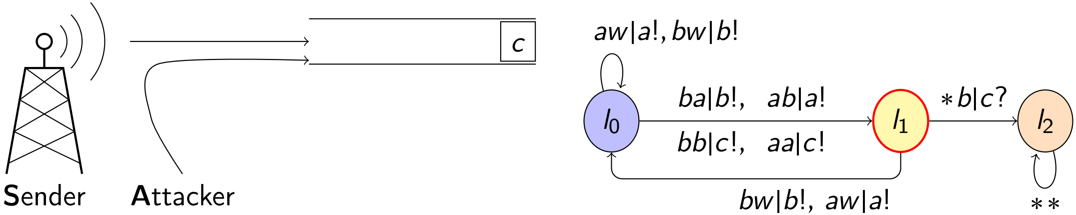




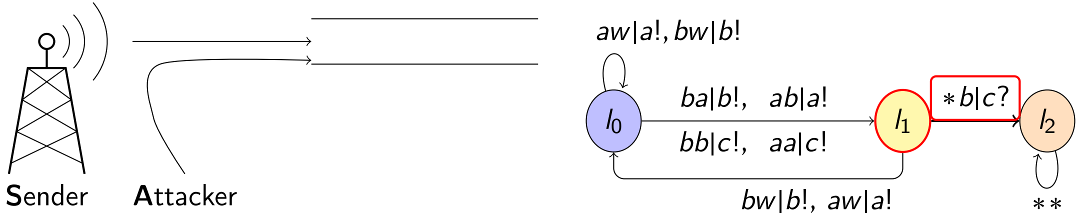
CSLCG: Def by Example



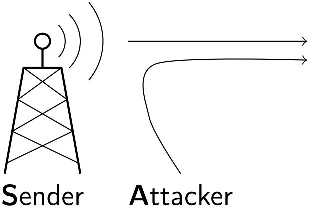
CSLCG: Def by Example

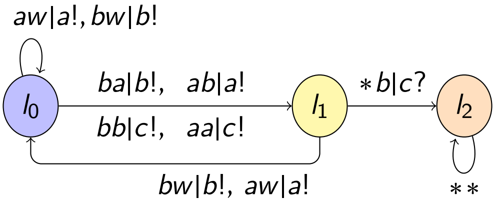


CSLCG: Def by Example



CSLCG: Def by Example





From $s =$ l $\cdot w$:

1. Pick an action for every player, then take the corresponding $l \xrightarrow{f} l'$
2. **Change** location to l' ;
3. Apply the **channel operation** f on w to get w';
4. Drop from w' to $w'' \preceq w'$: **subword ordering**.

Result: $s' =$ l' $\cdot w''$:

Contribution 1: Zero-sum case

The fixed point algorithms ([FS01, dAHK07]) still apply!

Contribution 1: Zero-sum case

The fixed point algorithms ([FS01, dAHK07]) still apply!

- $\mathrm{Pre}_i(X) = \{s \mid i$ can enforce reaching X in one step with pp.$\}$.
- Pre_i is computable for regular sets.
- **Termination**: Thanks to WQO's property [FS01].
- **Correctness**: Finite Attractor property [BBS06].

Contribution 1: Zero-sum case

The fixed point algorithms ([FS01, dAHK07]) still apply!

- $\mathrm{Pre}_i(X) = \{s \mid i$ can enforce reaching X in one step with pp.$\}$.
- Pre_i is computable for regular sets.
- **Termination**: Thanks to WQO's property [FS01].
- **Correctness**: Finite Attractor property [BBS06].

Theorem

*Let $R \subseteq L \cdot M^*$ a **regular** set of configurations. One can compute the set of winning configurations:*

- *Positive P. Reachability: $[\![\mathrm{NZ}(\Diamond R)]\!]_1$;*
- *Almost Sure Reachability: $[\![\mathrm{AS}(\Diamond R)]\!]_1$;*

Contribution 1: Zero-sum case

The fixed point algorithms ([FS01, dAHK07]) still apply!

- $\mathrm{Pre}_i(X) = \{s \mid i$ can enforce reaching X in one step with pp.$\}$.
- Pre_i is **computable** for regular sets.
- **Termination**: Thanks to **WQO's property** [FS01].
- **Correctness**: **Finite Attractor** property [BBS06].

Theorem

*Let $R \subseteq L \cdot M^*$ a **regular** set of configurations. One can compute the set of winning configurations:*

- ***Positive P.*** *Reachability:* $[\![\mathrm{NZ}(\Diamond R)]\!]_1$;
- ***Almost sure*** *Safety:* $[\![\mathrm{AS}(\Box R)]\!]_1$;
- ***Almost Sure*** *Reachability:* $[\![\mathrm{AS}(\Diamond R)]\!]_1$;
- ***Positive P.*** *Safety:* $[\![\mathrm{NZ}(\Box R)]\!]_1$;

Contribution 2: Conjunction of Objectives

Theorem

Let Φ *be a conjunction of* NZ *and* AS *objectives for safety and reachability path specifications. Then the winning region* $[\![\Phi]\!]_i$ *is computable.*

$\leadsto$ More in the paper: how to represent/combine winning strategies with possibly **infinite memory** (case $\mathrm{NZ}(\Box\ldots)$) with **infinite state space**.

Contribution 2: Conjunction of Objectives

Theorem

Let Φ be a conjunction of NZ and AS objectives for safety and reachability path specifications. Then the winning region $[\![\Phi]\!]_i$ is computable.

$\rightsquigarrow$ More in the paper: how to represent/combine winning strategies with possibly **infinite memory** (case $\mathrm{NZ}(\Box\ldots)$) with **infinite state space**.

NB: It is the "maximal" possible result:

-  Bertrand and al [BBS07] proves that $\mathrm{NZ}(\Box\Diamond R)$ (Büchi) and $\mathrm{AS}(\Box\Diamond R_1 \wedge \Diamond\Box R_2)$ cases are undecidable.
-  [May03] proved that $[\![\mathrm{E}(\Box R)]\!]_1$ cannot be computed.

Contribution 3: Core

Theorem

For a pair $(\mathscr{G}, \Gamma)$ *where players' objectives are almost-sure reachability or almost-sure safety objectives, and property* $\Gamma = \mathrm{AS}(\varphi)$ *with* φ *of the form* $\bigwedge_i \Diamond R_i$, $\bigwedge_i \Box R_i$, *or* $\bigwedge_i \Box\Diamond R_i$, *the problems of* E-Core *and* A-Core *are* <u>*decidable*</u>*.*

Contribution 3: Core

Theorem

For a pair $(\mathcal{G},\Gamma)$ where players' objectives are almost-sure reachability or almost-sure safety objectives, and property $\Gamma = \mathrm{AS}(\varphi)$ with φ of the form $\bigwedge_i \Diamond R_i$, $\bigwedge_i \Box R_i$, or $\bigwedge_i \Box\Diamond R_i$, the problems of E-Core *and* A-Core *are* decidable.

- Guess the set of winning players $W \subseteq \mathrm{Agt}$;
- Check that the 1.5-player game of Agt vs $\emptyset$ is **winning for the conjunction**:

$$\Gamma \wedge \bigwedge_{i\in W} \mathrm{AS}(\varphi_i) \wedge \bigwedge_{i\notin W} \mathrm{NZ}(\neg\varphi_i)$$

- For all $C \subseteq \overline{W}$, check that C against $\overline{C}$ is losing.

Contribution 3: Core

Theorem

For a pair $(\mathcal{G}, \Gamma)$ where players' objectives are almost-sure reachability or almost-sure safety objectives, and property $\Gamma = \mathrm{AS}(\varphi)$ with φ of the form $\bigwedge_i \Diamond R_i$, $\bigwedge_i \Box R_i$, or $\bigwedge_i \Box\Diamond R_i$, the problems of E-Core *and* A-Core *are* <u>*decidable*</u>*.*

- Guess the set of winning players $W \subseteq \mathrm{Agt}$;
- Check that the 1.5-player game of Agt vs $\emptyset$ is **winning for the conjunction**:

$$\Gamma \wedge \bigwedge_{i \in W} \mathrm{AS}(\varphi_i) \wedge \bigwedge_{i \notin W} \mathrm{NZ}(\neg\varphi_i)$$

- For all $C \subseteq \overline{W}$, check that C against $\overline{C}$ is **losing**.

NB: "maximal" result since Büchi+coBüchi objectives make the problem undecidable [BBS07].

Summary and future work

- Rational Verification problem on infinite state still decidable;
- Qualitative Objectives for Stochastic Equilibria;
-  vs : The probabilistic setting is more computable than the ND one.

Summary and future work

- Rational Verification problem on infinite state still decidable;
- Qualitative Objectives for Stochastic Equilibria;
-  vs : The probabilistic setting is more computable than the ND one.

Future work:

- Restrictions on the strategy classes (FM only?);
- Nash Equilibria;
- Partial observation (Signal Games).

Summary and future work

- Rational Verification problem on infinite state still **decidable**;
- Qualitative Objectives for Stochastic Equilibria;
-  vs : The probabilistic setting is more computable than the ND one.

Future work:

- Restrictions on the strategy classes (FM only?);
- Nash Equilibria;
- Partial observation (Signal Games).

Thank you for your attention

Bibliography I

[AABN18] Parosh Aziz Abdulla, Mohamed Faouzi Atig, Ahmed Bouajjani, and Tuan Phong Ngo. A load-buffer semantics for total store ordering. Logical Methods in Computer Science, 14(1), 2018.

[BBS06] Christel Baier, Nathalie Bertrand, and Philippe Schnoebelen. A note on the attractor-property of infinite-state markov chains. Inf. Process. Lett., 97(2):58–63, 2006.

[BBS07] Christel Baier, Nathalie Bertrand, and Philippe Schnoebelen. Verifying nondeterministic probabilistic channel systems against ω-regular linear-time properties. ACM Trans. Comput. Log., 9(1):5, 2007.

[dAHK07] Luca de Alfaro, Thomas A. Henzinger, and Orna Kupferman. Concurrent reachability games. Theor. Comput. Sci., 386(3):188–217, 2007.

[FS01] Alain Finkel and Philippe Schnoebelen. Well-structured transition systems everywhere! Theor. Comput. Sci., 256(1-2):63–92, 2001.

[May03] Richard Mayr. Undecidable problems in unreliable computations. Theor. Comput. Sci., 297(1-3):337–354, 2003.

Strategy Classes

A strategy for player i is:

$$\sigma_i : S^+ \to \mathrm{Dist}(\mathrm{Act})$$

- **D**etermistic: only one action with probability 1;

$$\forall h \in S^+, \exists \alpha : \sigma_i(h)[\alpha] = 1$$

Strategy Classes

A strategy for player i is:

$$\sigma_i : S^+ \to \mathrm{Dist}(\mathrm{Act})$$

- **D**etermistic: only one action with probability 1;

$$\forall h \in S^+, \exists \alpha : \sigma_i(h)[\alpha] = 1$$

- **P**ositional: depends only on the current state;

$$\forall h \in S^*, \forall s \in S^+, \sigma_i(h \cdot s) = \sigma_i(s)$$

Strategy Classes

A strategy for player i is:

$$\sigma_i : S^+ \rightarrow \mathrm{Dist}(\mathrm{Act})$$

- **D**etermistic: only one action with probability 1;

$$\forall h \in S^+, \exists \alpha : \sigma_i(h)[\alpha] = 1$$

- **P**ositional: depends only on the current state;

$$\forall h \in S^*, \forall s \in S^+, \sigma_i(h \cdot s) = \sigma_i(s)$$

Strategy Classes

A strategy for player i is:

$$\sigma_i : S^+ \to \mathrm{Dist}(\mathrm{Act})$$

- **D**etermistic: only one action with probability 1;

$$\forall h \in S^+, \exists \alpha : \sigma_i(h)[\alpha] = 1$$

- **P**ositional: depends only on the current state;

$$\forall h \in S^*, \forall s \in S^+, \sigma_i(h \cdot s) = \sigma_i(s)$$

Can't we just play with **DP** strategies only?

Strategy Classes, Updated

A strategy for player i is: $\sigma_i : (L \cdot M^*)^+ \to \mathrm{Dist}(\mathrm{Act})$

- **D**etermistic: only one action with probability 1;

$$\forall h \in (L \cdot M^*)^+, \exists \alpha : \sigma_i(h)[\alpha] = 1$$

Strategy Classes, Updated

A strategy for player i is: $\sigma_i : (L \cdot M^*)^+ \to \mathrm{Dist}(\mathrm{Act})$

- **D**etermistic: only one action with probability 1;

$$\forall h \in (L \cdot M^*)^+, \exists \alpha : \sigma_i(h)[\alpha] = 1$$

- **P**ositional: depends only on the current state;

$$\forall h \in (L \cdot M^*)^*, \forall s \in L \cdot M^*, \sigma_i(h \cdot s) = \sigma_i(s)$$

Strategy Classes, Updated

A strategy for player i is: $\sigma_i : (L \cdot M^*)^+ \to \mathrm{Dist}(\mathrm{Act})$

- **D**etermistic: only one action with probability 1;

$$\forall h \in (L \cdot M^*)^+, \exists \alpha : \sigma_i(h)[\alpha] = 1$$

- **P**ositional: depends only on the current state;

$$\forall h \in (L \cdot M^*)^*, \forall s \in L \cdot M^*, \sigma_i(h \cdot s) = \sigma_i(s)$$

- **F**inite **M**emory: the distribution of actions can be computed by a finite automaton.

$$\forall \delta \in \mathrm{Dist}(\mathrm{Act}), \{h \in (L \cdot M^*)^+ \mid \sigma_i(h) = \delta\} \text{ is a regular set}$$

and the set of possible distributions is **finite**.

Strategy Classes, Updated

A strategy for player i is: $\sigma_i : (L \cdot M^*)^+ \to \mathrm{Dist}(\mathrm{Act})$

- **D**etermistic: only one action with probability 1;

$$\forall h \in (L \cdot M^*)^+, \exists \alpha : \sigma_i(h)[\alpha] = 1$$

- **P**ositional: depends only on the current state;

$$\forall h \in (L \cdot M^*)^*, \forall s \in L \cdot M^*, \sigma_i(h \cdot s) = \sigma_i(s)$$

- **F**inite **M**emory: the distribution of actions can be computed by a finite automaton.

$$\forall \delta \in \mathrm{Dist}(\mathrm{Act}), \{h \in (L \cdot M^*)^+ \mid \sigma_i(h) = \delta\} \text{ is a regular set}$$

  and the set of possible distributions is **finite**.

P strategies **may** not be finitely represented. **PFM** are finitely represented, **Counting** too.

Strategy Classes, Updated, Counting

A strategy for player i is **C**ounting: if there exist two PFM strategies σ^u, σ^v such that:

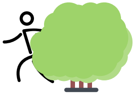
$$\forall n \geq 1, \forall h \in S^n, \sigma(h) = p_n \cdot \sigma^u(h) + (1 - p_n)\sigma^v(h)$$

Where:

$$p_n = 2^{-1/(2^k)}$$

Counting strategies are sufficient for winning $\mathrm{NZ}(\Box \cdots)$.

Skirmish Game [dAHK07]

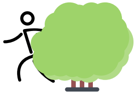


$A_1(s_0) = \{h, r\}$

$A_2(s_0) = \{s, w\}$

Played:

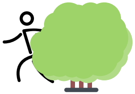
Skirmish Game [dAHK07]



$A_1(s_0) = \{h, r\}$ $A_2(s_0) = \{s, w\}$

Played: hw

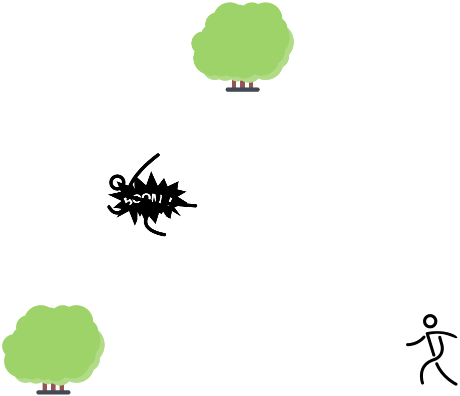
Skirmish Game [dAHK07]



$A_1(s_0) = \{h, r\}$ $A_2(s_0) = \{s, w\}$

Played: rs

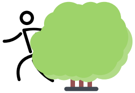
Skirmish Game [dAHK07]



$A_1(s_0) = \{h, r\}$ $A_2(s_0) = \{s, w\}$

Played: rs

Skirmish Game [dAHK07]

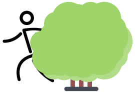


$A_1(s_0) = \{h, r\}$

$A_2(s_0) = \{s, w\}$

Played: rs

Skirmish Game [dAHK07]

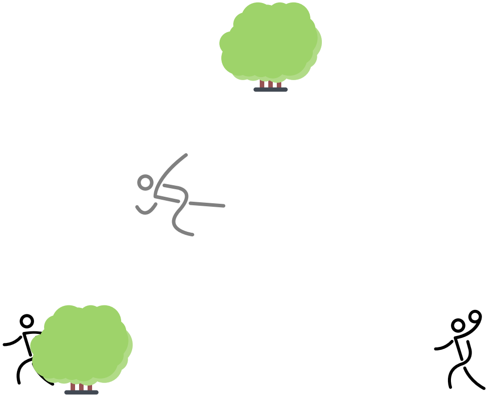


$A_1(s_0) = \{h, r\}$

$A_2(s_0) = \{s, w\}$

Played: rw

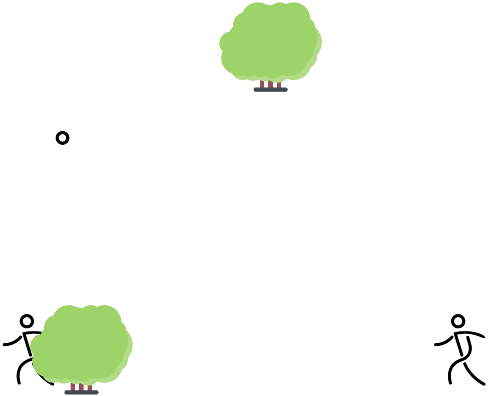
Skirmish Game [dAHK07]



$A_1(s_0) = \{h, r\}$ $A_2(s_0) = \{s, w\}$

Played: rw

Skirmish Game [dAHK07]

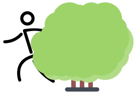


$A_1(s_0) = \{h, r\}$

$A_2(s_0) = \{s, w\}$

Played: rw

Skirmish Game [dAHK07]

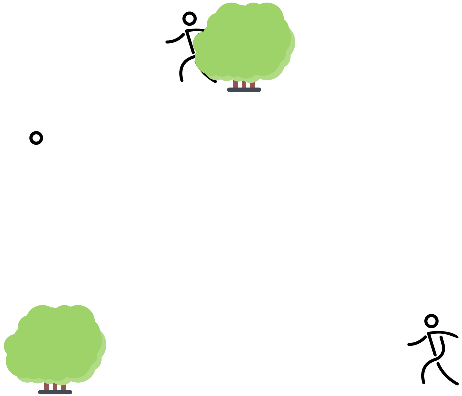


$A_1(s_0) = \{h, r\}$

$A_2(s_0) = \{s, w\}$

Played: hs

Skirmish Game [dAHK07]

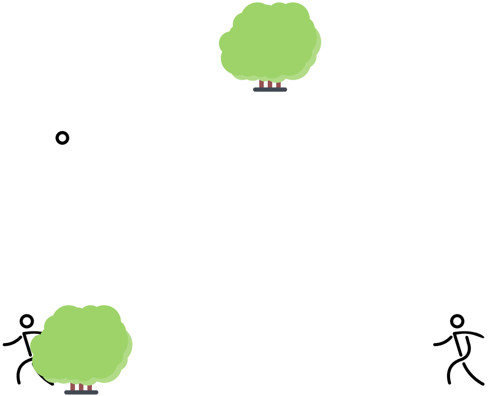


$A_1(s_0) = \{h, r\}$

$A_2(s_0) = \{s, w\}$

Played: hs

Skirmish Game [dAHK07]



$A_1(s_0) = \{h, r\}$

$A_2(s_0) = \{s, w\}$

Played: hs