Life is Random, Time is Not

Markov Decision Processes with Window Objectives

Thomas Brihaye (UMONS)
Youssouf Oualhadj (LACL – UPEC)
Florent Delgrange (UMONS & RWTH)
Mickael Randour (UMONS & FNRS)

May 29, 2019

UnRAVeL Seminar, RWTH Aachen University

Research supported by F.R.S.-FNRS (ManySynth)

Strategy synthesis

Finding good controllers for systems interacting with an environment

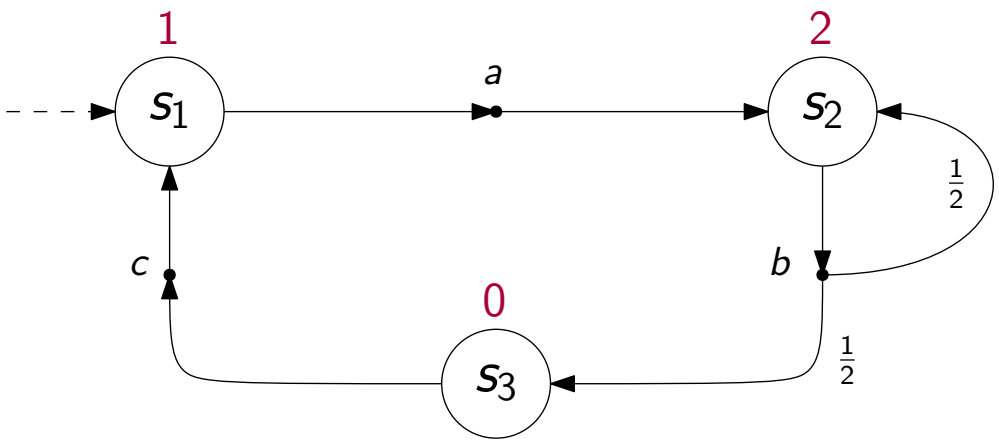
- **Game setting:** ensure a specified behavior against all possible strategies of the environment
- **Markov Decision Process (MDP) setting:**
  - environment **stochastic**
  - ensure a specified behavior with a sufficient probability
- **Classical objectives** reason about infinite runs *in their limit*
- **Window objectives in games** [CDRR15, BHR16] : ***ensure a good behavior in a parametrized time frame all along the run***
  - ⇝ *conservative approximations of classical objectives*

Aim of this talk

Introducing **window objectives** in the **stochastic** context

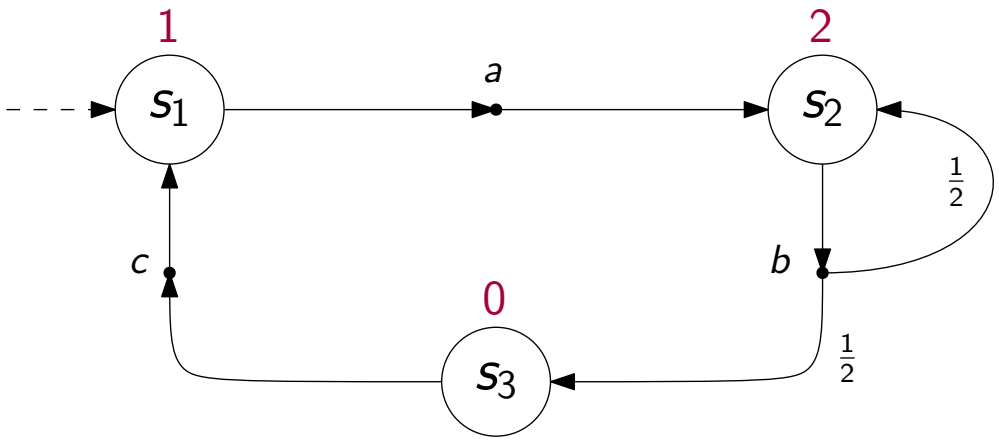
Example

- **Parity:** asks the **minimum priority seen infinitely often to be even**
  - ⇝ canonical way of encoding ω-regular properties
- ⇝ controller winning



Example

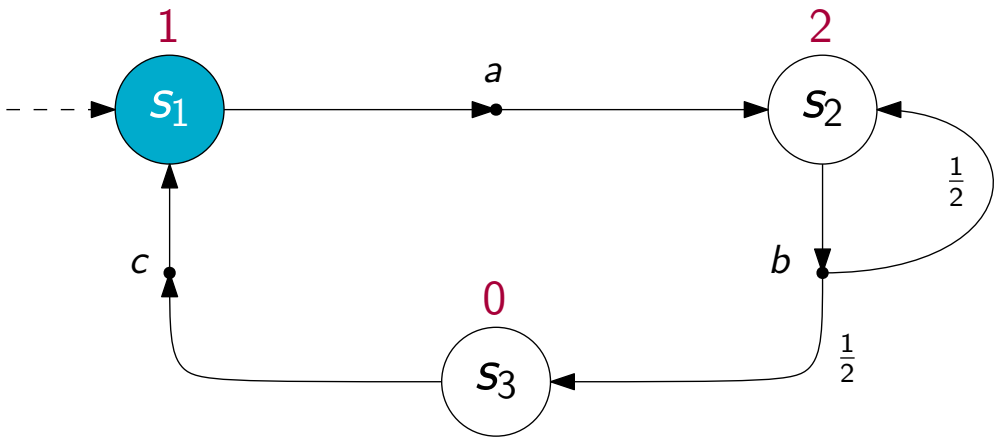
- **Parity:** asks the **minimum priority seen infinitely often to be even**
  - ⇝ canonical way of encoding ω-regular properties
- ⇝ controller winning



- **Window parity:** asks **the minimum priority seen within at most $\lambda > 0$ time steps to be even from each position of the infinite run**

Example

- **Parity:** asks the **minimum priority seen infinitely often to be even**
  - ⇝ canonical way of encoding ω-regular properties
- ⇝ controller winning

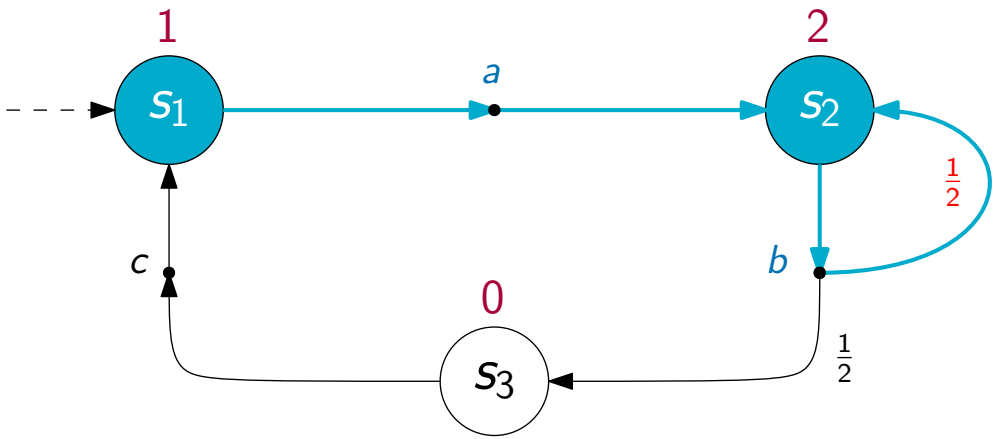


- **Window parity:** asks **the minimum priority seen within at most $\lambda > 0$ time steps to be even from each position of the infinite run**
- Every time s_1 is visited, there is a probability > 0 of not seeing the priority 0 before λ steps ($\frac{1}{2^{\lambda-1}}$)

Example

- **Parity:** asks the **minimum priority seen infinitely often to be even**
  - ⇝ canonical way of encoding ω-regular properties

⇝ controller winning

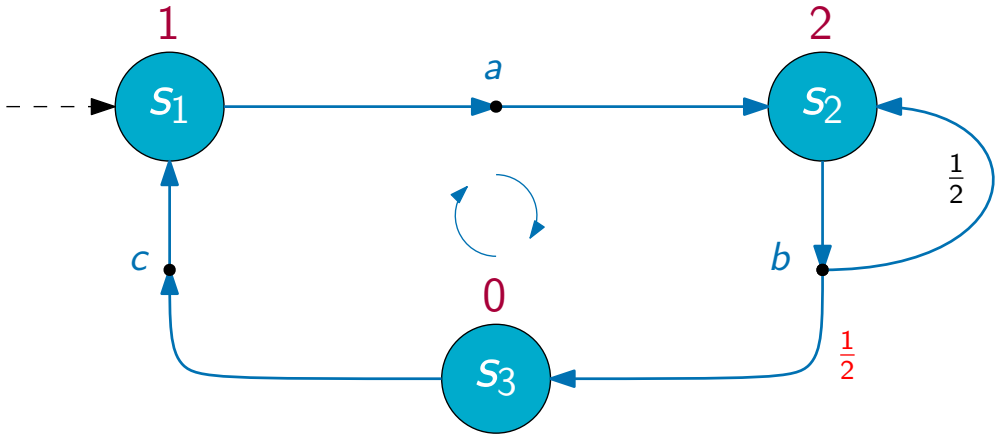


- **Window parity:** asks **the minimum priority seen within at most $\lambda > 0$ time steps to be even from each position of the infinite run**
- Every time s_1 is visited, there is a probability > 0 of not seeing the priority 0 before λ steps ($\frac{1}{2^{\lambda-1}}$)

Example

- **Parity:** asks the **minimum priority seen infinitely often to be even**
  - ⇝ canonical way of encoding ω-regular properties

⇝ controller winning



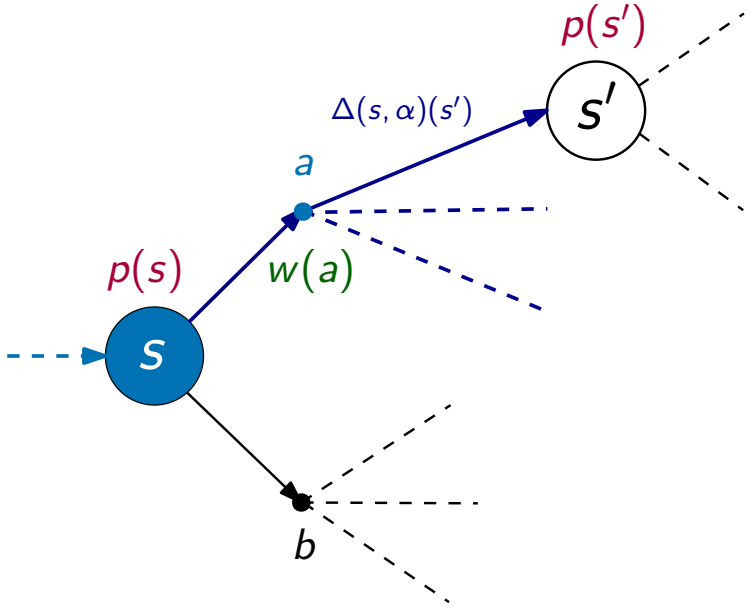
- **Window parity:** asks **the minimum priority seen within at most $\lambda > 0$ time steps to be even from each position of the infinite run**
- Every time s_1 is visited, there is a probability > 0 of not seeing the priority 0 before λ steps ($\frac{1}{2^{\lambda-1}}$)

⇝ probability zero

Outline

1. Model
2. Objectives
   - 2.1 Classical long-run objectives
   - 2.3 Window objectives
   - 2.5 Decision problem
3. Fixed case
   - 3.1 Reductions
   - 3.3 Direct Fixed Window
4. Prefix independent objectives
   - 4.1 The case of end-components
   - 4.3 Fixed and Bounded window
5. Extensions

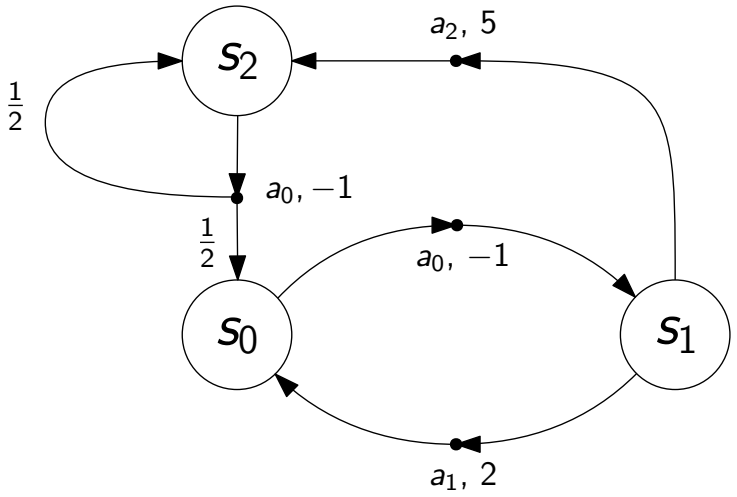
Markov Decision Process (MDP)



An MDP $\mathcal{M} = (S, A, \Delta)$ is a tuple such that

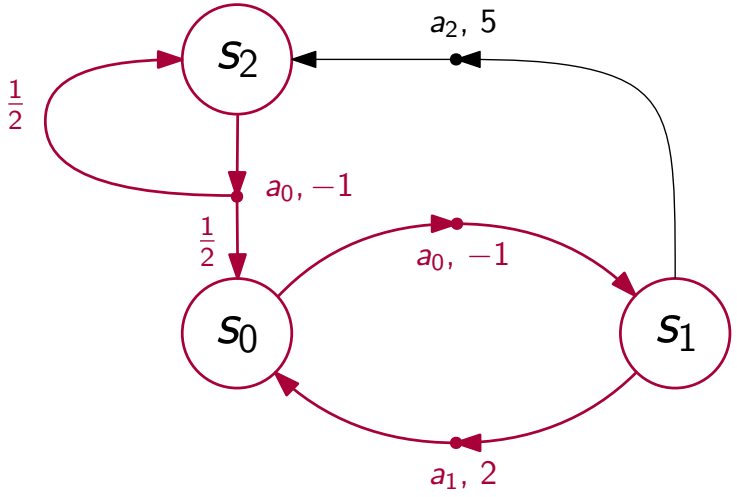
- S is the **set of states** of the system
- A is the **set of actions** of the system
- $\Delta : S \times A \to \mathcal{D}(S)$ is the **probability transition function**
- $w : A \to \mathbb{Z}$ is a **weight function**
- $p : S \to \{0, 1, \ldots, d\}$ is a **priority function** ($d \leq |S| + 1$ w.l.o.g.)

Runs and strategies



- **Runs**: $\rho = s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} s_2 \ldots s_n \xrightarrow{a_n} \cdots \in \mathrm{Runs}(\mathcal{M})$ such that $\Delta(s_i, a_i)(s_{i+1}) > 0$
- **Strategy**: σ chooses at each step an action
  - **pure finite-memory strategies**: choose actions according to a finite number of informations gathered in the past
  - **pure memoryless strategies**: $\sigma : S \to A$

Runs and strategies

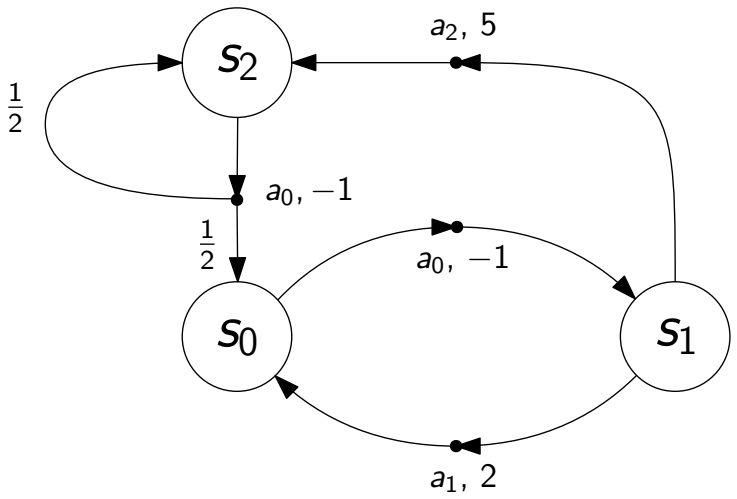


- **Fix a strategy** σ
- $\rho \in \mathrm{Runs}^{\sigma}(\mathcal{M})$ ⇝ runs compatible with σ
- Induce a ***discrete-time Markov chain***: fully stochastic process $\mathcal{M}^{\sigma}$
  - ⇝ **Event**: $E \subseteq \mathrm{Runs}^{\sigma}(\mathcal{M})$
  - ⇝ $\mathbb{P}^{\sigma}_{\mathcal{M}}[E]$: probability measure of the event E

Outline

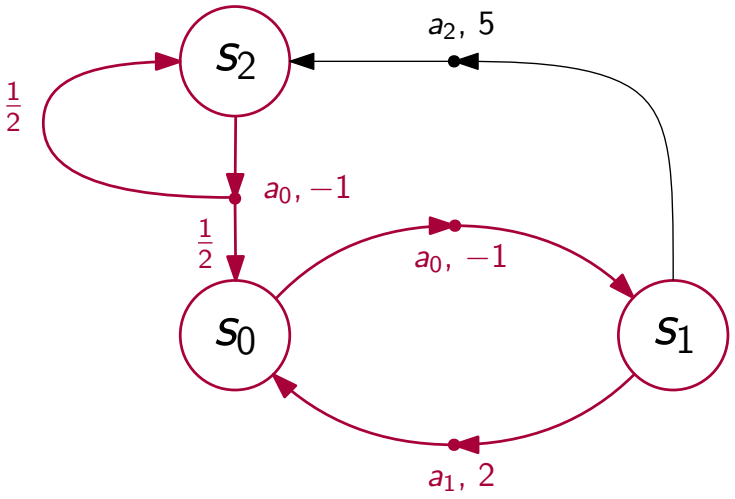
1. Model
2. Objectives
   - 2.1 Classical long-run objectives
   - 2.3 Window objectives
   - 2.5 Decision problem
3. Fixed case
   - 3.1 Reductions
   - 3.3 Direct Fixed Window
4. Prefix independent objectives
   - 4.1 The case of end-components
   - 4.3 Fixed and Bounded window
5. Extensions

Mean-Payoff objective



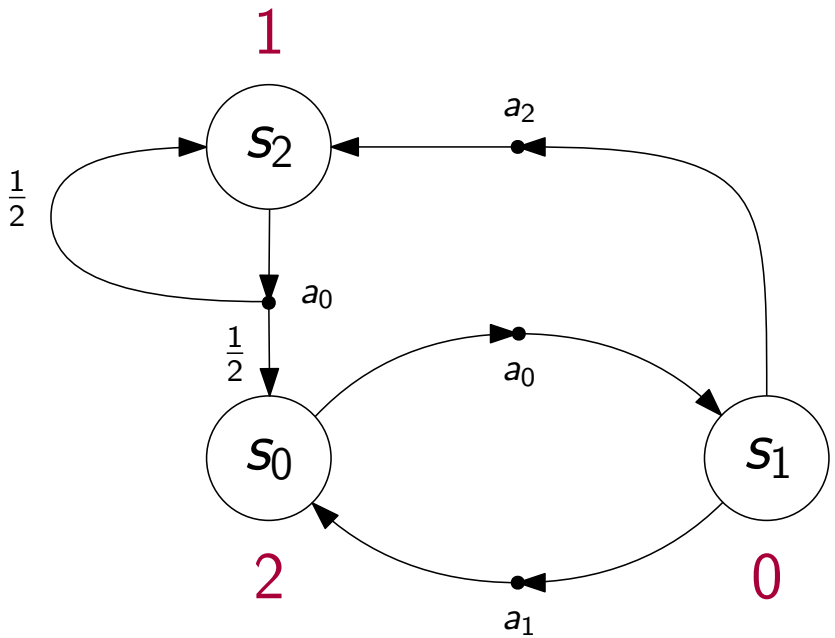
- $\mathrm{MP}(\rho = s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} \dots) = \liminf_{n\to\infty} \frac{1}{n} \cdot \sum_{i=0}^{n} a_i$
  - **Example**: $\forall n \in \mathbb{N}$,
    $\mathrm{MP}\big((s_2 \xrightarrow{a_0})^n (s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1})^\omega\big) = \mathrm{MP}\big((s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1})^\omega\big) = \frac{1}{2}$
- $\mathrm{MeanPayoff} = \{\rho \in \mathrm{Runs}(\mathcal{M}) \mid \mathrm{MP}(\rho) \geq 0\}$

Mean-Payoff objective



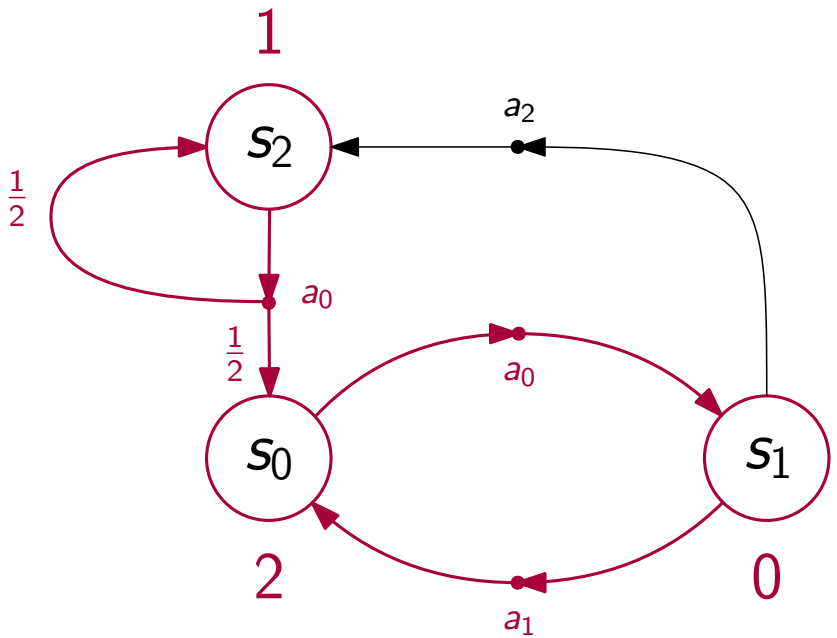
- $\mathrm{MP}(\rho = s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} \dots) = \liminf_{n\to\infty} \frac{1}{n} \cdot \sum_{i=0}^{n} a_i$
  - **Example**: $\forall n \in \mathbb{N}$,
    $\mathrm{MP}\big((s_2 \xrightarrow{a_0})^n (s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1})^\omega\big) = \mathrm{MP}\big((s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1})^\omega\big) = \frac{1}{2}$
- $\mathrm{MeanPayoff} = \{\rho \in \mathrm{Runs}(\mathcal{M}) \mid \mathrm{MP}(\rho) \geq 0\}$
- $\mathbb{P}^{\sigma}_{\mathcal{M},s_2}[\mathrm{MeanPayoff}] = 1$

Parity objective



- Parity $= \{\rho \in \mathrm{Runs}(\mathcal{M}) \mid \min_{s \in \inf(\rho)} p(s) = 0 \pmod 2\}$

Parity objective



- $\text{Parity} = \{\rho \in \text{Runs}(\mathcal{M}) \mid \min_{s \in \inf(\rho)} p(s) = 0 \pmod 2\}$
- $\mathbb{P}^{\sigma}_{\mathcal{M},s_2}[\text{Parity}] = 1$

Window Objectives

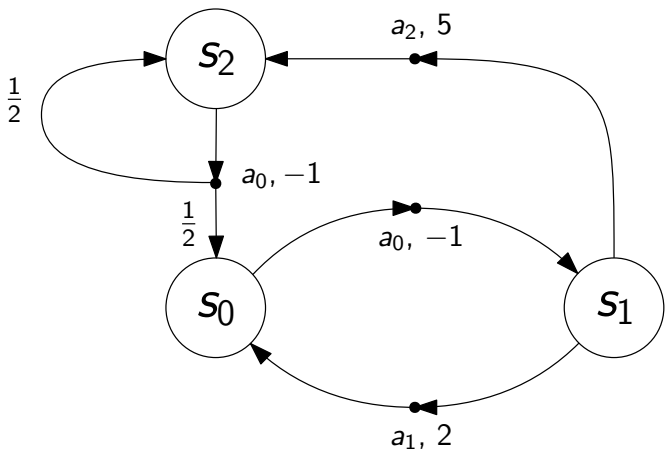
- Runs that exhibit **good behaviors** within a configurable **time frame**
- **Strengthen traditional objectives** (correct behaviors at the limit)

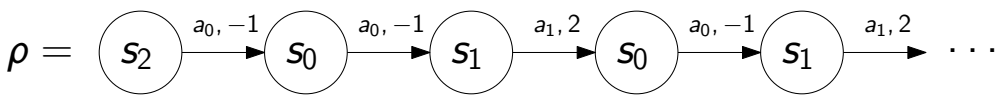
⇝ Make use of the **window formalism** to reason about **behaviors** in a given time bound $\lambda > 0$.

$$\mathrm{GW}(\lambda) = \{\rho \in \mathrm{Runs}(\mathcal{M}) \mid \textit{good behavior in at most } \lambda \textit{ steps from } s_0\}$$

- Positive sum in at most $\lambda = 3$ steps?

Window Mean-Payoff





Window Objectives

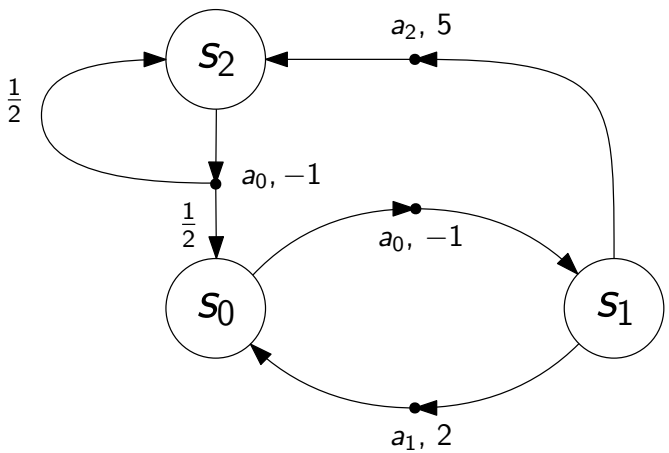
- Runs that exhibit **good behaviors** within a configurable **time frame**
- **Strengthen traditional objectives** (correct behaviors at the limit)

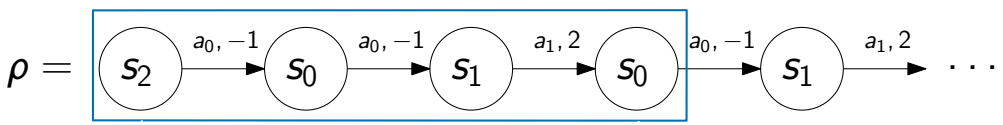
⇝ Make use of the **window formalism** to reason about **behaviors** in a given time bound $\lambda > 0$.

$$\mathrm{GW}(\lambda) = \{\rho \in \mathrm{Runs}(\mathcal{M}) \mid \textit{good behavior in at most } \lambda \textit{ steps from } s_0\}$$

Window Mean-Payoff

- Positive sum in at most $\lambda = 3$ steps?
- Window of maximal size $\lambda = 3$





Window Objectives

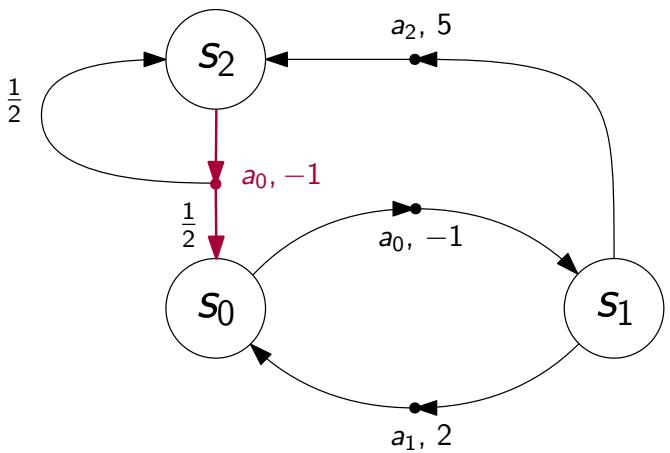
- Runs that exhibit **good behaviors** within a configurable **time frame**
- **Strengthen traditional objectives** (correct behaviors at the limit)

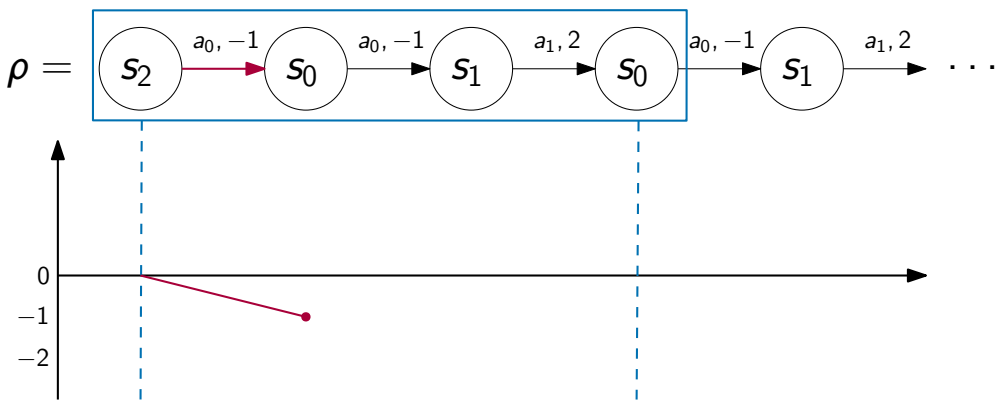
⇝ Make use of the **window formalism** to reason about **behaviors** in a given time bound $\lambda > 0$.

$$\mathrm{GW}(\lambda) = \{\rho \in \mathrm{Runs}(\mathcal{M}) \mid \textit{good behavior in at most } \lambda \textit{ steps from } s_0\}$$

Window Mean-Payoff

- Positive sum in at most $\lambda = 3$ steps?
- Window of maximal size $\lambda = 3$





Window Objectives

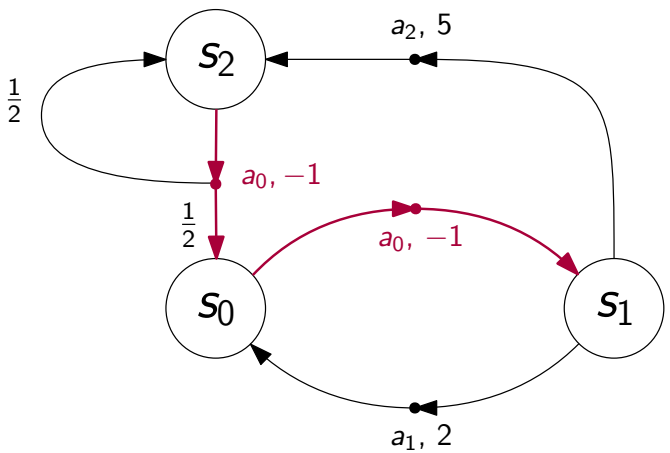
- Runs that exhibit **good behaviors** within a configurable **time frame**
- **Strengthen traditional objectives** (correct behaviors at the limit)

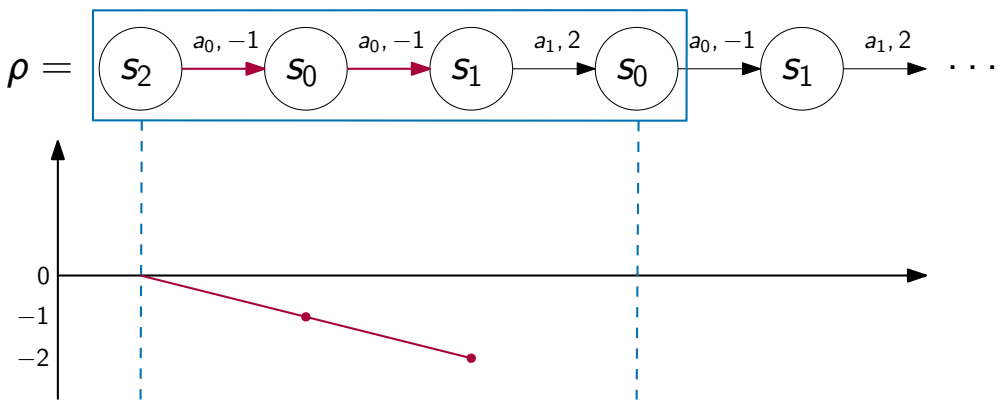
⇝ Make use of the **window formalism** to reason about **behaviors** in a given time bound $\lambda > 0$.

$$\mathrm{GW}(\lambda) = \{\rho \in \mathrm{Runs}(\mathcal{M}) \mid \textit{good behavior in at most } \lambda \textit{ steps from } s_0\}$$

Window Mean-Payoff

- Positive sum in at most $\lambda = 3$ steps?
- Window of maximal size $\lambda = 3$





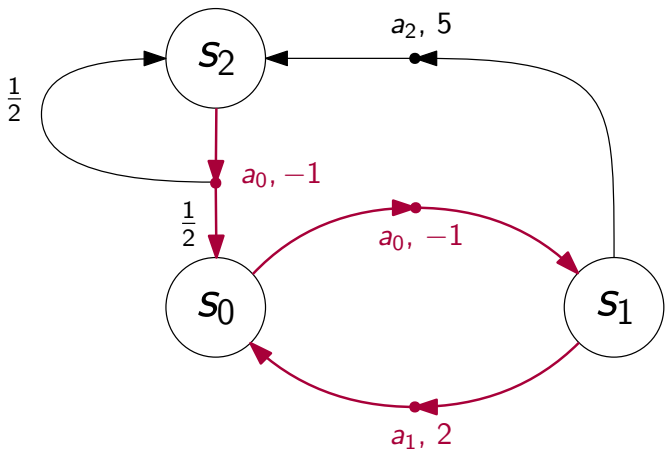
Window Objectives

- Runs that exhibit **good behaviors** within a configurable **time frame**
- **Strengthen traditional objectives** (correct behaviors at the limit)

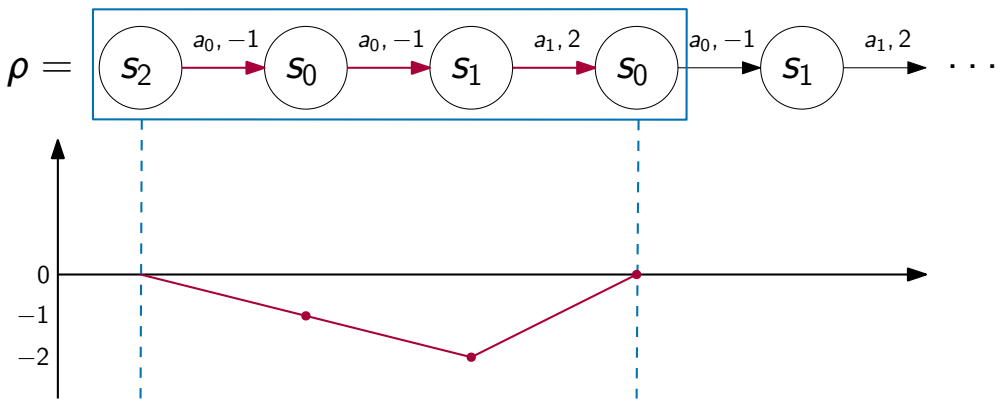
⇝ Make use of the **window formalism** to reason about **behaviors** in a given time bound $\lambda > 0$.

$$\mathrm{GW}(\lambda) = \{\rho \in \mathrm{Runs}(\mathcal{M}) \mid \textit{good behavior in at most } \lambda \textit{ steps from } s_0\}$$

Window Mean-Payoff



- Positive sum in at most $\lambda = 3$ steps?
- Window of maximal size $\lambda = 3$



⇝ Good window of size $\lambda = 3$

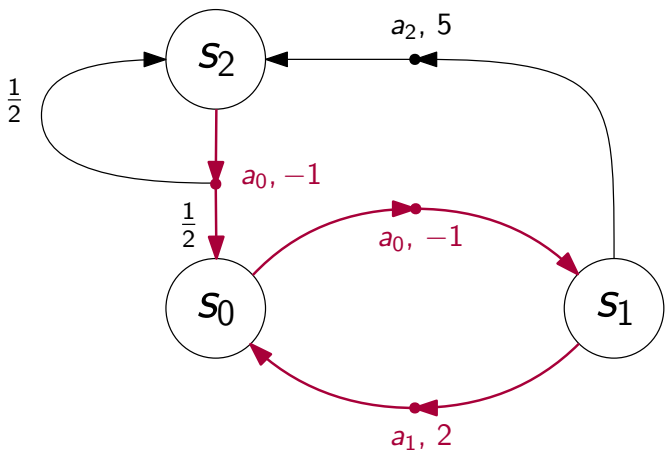
Window Objectives

- Runs that exhibit **good behaviors** within a configurable **time frame**
- **Strengthen traditional objectives** (correct behaviors at the limit)

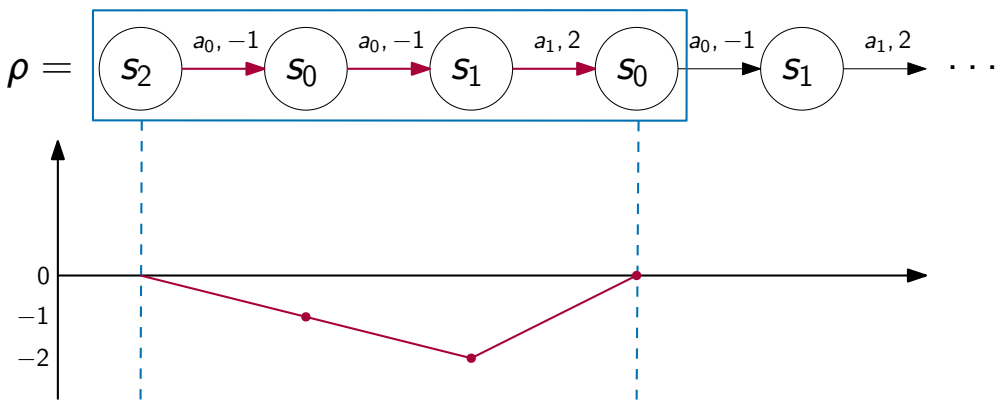
⇝ Make use of the **window formalism** to reason about **behaviors** in a given time bound $\lambda > 0$.

$$\mathrm{GW}(\lambda) = \{\rho \in \mathrm{Runs}(\mathcal{M}) \mid \textit{good behavior in at most } \lambda \textit{ steps from } s_0\}$$

Window Mean-Payoff



- Positive sum in at most $\lambda = 3$ steps?
- Window of maximal size $\lambda = 3$



⇝ Good window of size $\lambda = 3$

⇝ $\rho \in \mathrm{GW}_{\mathrm{mp}}(3)$

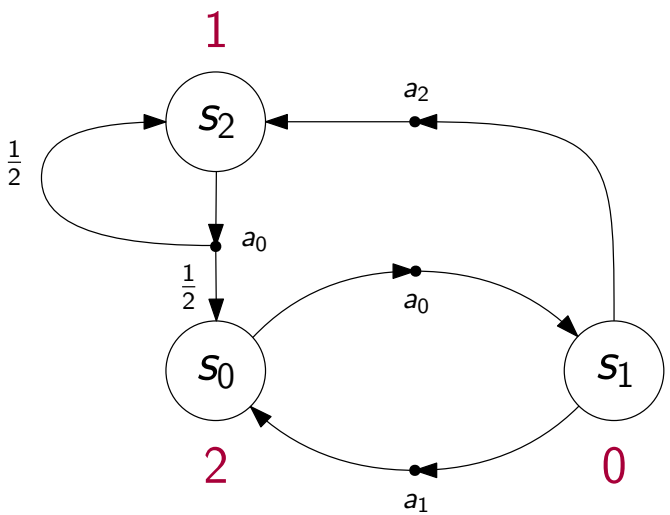
Window Objectives

- Runs that exhibit **good behaviors** within a configurable **time frame**
- **Strengthen traditional objectives** (correct behaviors at the limit)

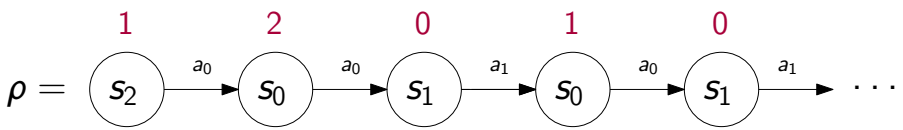
⇝ Make use of the **window formalism** to reason about **behaviors** in a given time bound $\lambda > 0$.

$$\mathrm{GW}(\lambda) = \{\rho \in \mathrm{Runs}(\mathcal{M}) \mid \textit{good behavior in at most } \lambda \textit{ steps from } s_0\}$$

Window Parity



- Minimum priority is even in at most $\lambda = 3$ steps?



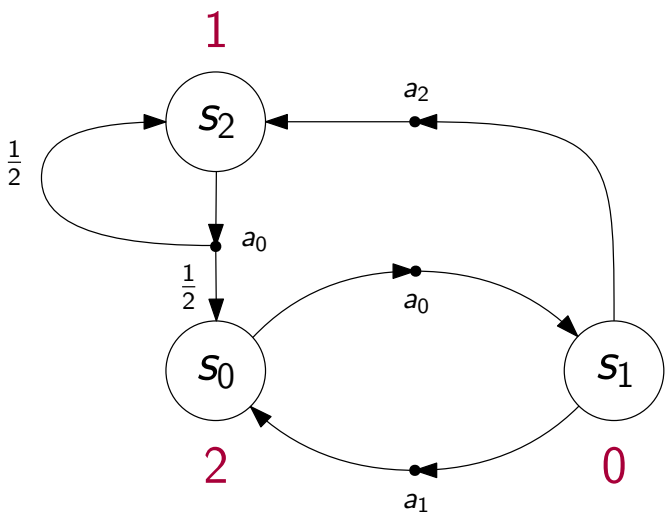
Window Objectives

- Runs that exhibit **good behaviors** within a configurable **time frame**
- **Strengthen traditional objectives** (correct behaviors at the limit)

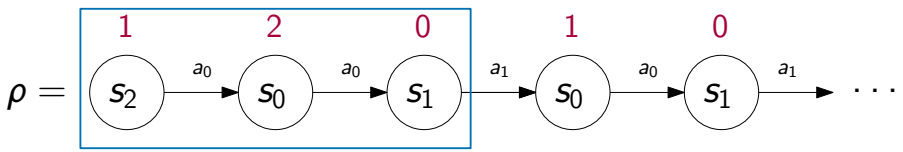
⇝ Make use of the **window formalism** to reason about **behaviors** in a given time bound $\lambda > 0$.

$$\mathrm{GW}(\lambda) = \{\rho \in \mathrm{Runs}(\mathcal{M}) \mid \textit{good behavior in at most } \lambda \textit{ steps from } s_0\}$$

Window Parity



- Minimum priority is even in at most $\lambda = 3$ steps?
- Window of maximal size $\lambda = 3$



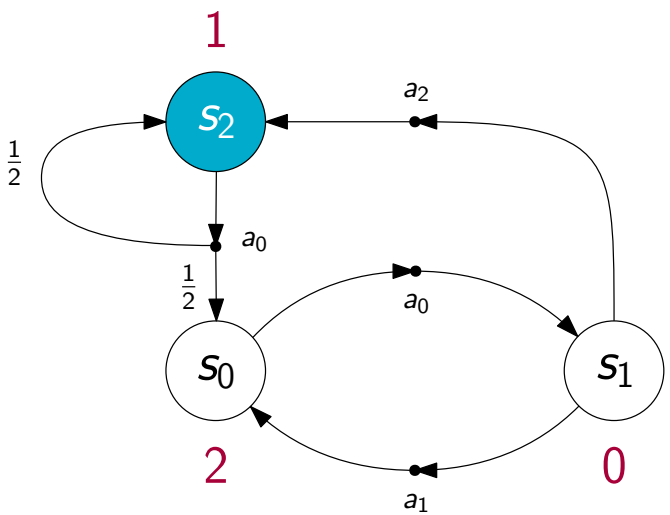
Window Objectives

- Runs that exhibit **good behaviors** within a configurable **time frame**
- **Strengthen traditional objectives** (correct behaviors at the limit)

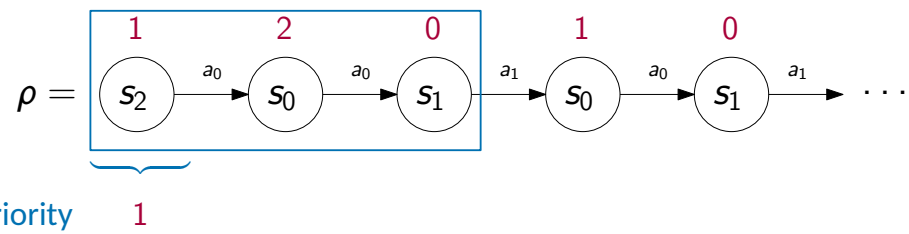
⇝ Make use of the **window formalism** to reason about **behaviors** in a given time bound $\lambda > 0$.

$$\mathrm{GW}(\lambda) = \{\rho \in \mathrm{Runs}(\mathcal{M}) \mid \textit{good behavior in at most } \lambda \textit{ steps from } s_0\}$$

Window Parity



- Minimum priority is even in at most $\lambda = 3$ steps?
- Window of maximal size $\lambda = 3$



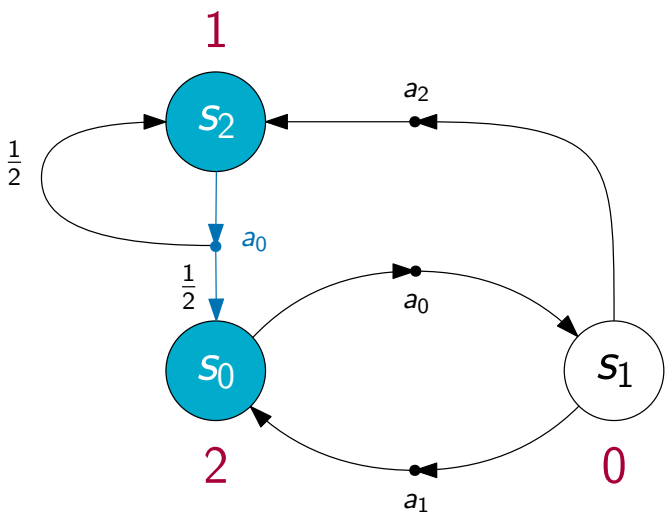
Window Objectives

- Runs that exhibit **good behaviors** within a configurable **time frame**
- **Strengthen traditional objectives** (correct behaviors at the limit)

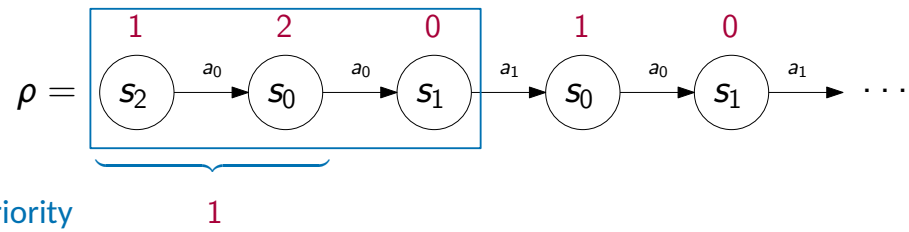
⇝ Make use of the **window formalism** to reason about **behaviors** in a given time bound $\lambda > 0$.

$$\mathrm{GW}(\lambda) = \{\rho \in \mathrm{Runs}(\mathcal{M}) \mid \textit{good behavior in at most } \lambda \textit{ steps from } s_0\}$$

Window Parity



- Minimum priority is even in at most $\lambda = 3$ steps?
- Window of maximal size $\lambda = 3$



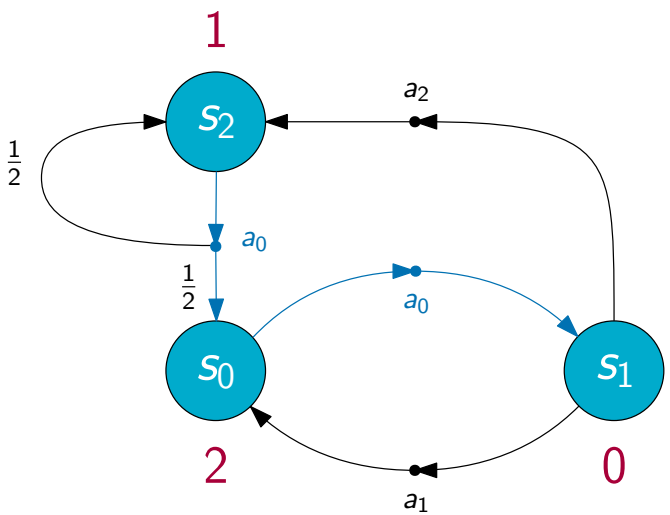
Window Objectives

- Runs that exhibit **good behaviors** within a configurable **time frame**
- **Strengthen traditional objectives** (correct behaviors at the limit)

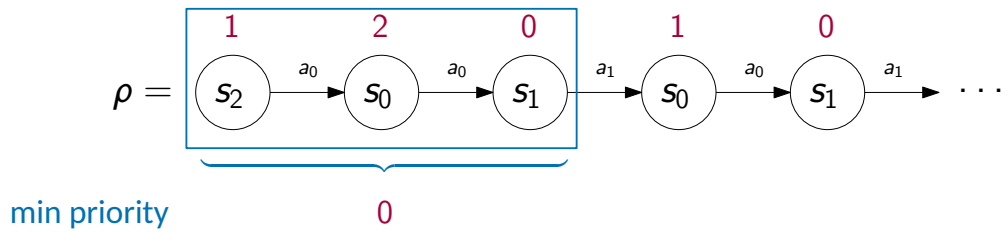
⇝ Make use of the **window formalism** to reason about **behaviors** in a given time bound $\lambda > 0$.

$$\mathrm{GW}(\lambda) = \{\rho \in \mathrm{Runs}(\mathcal{M}) \mid \textit{good behavior in at most } \lambda \textit{ steps from } s_0\}$$

Window Parity



- Minimum priority is even in at most $\lambda = 3$ steps?
- Window of maximal size $\lambda = 3$



⇝ Good window of size $\lambda = 3$

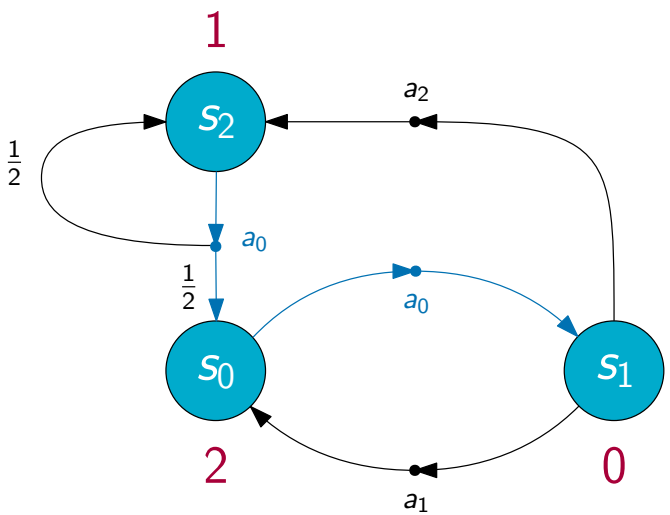
Window Objectives

- Runs that exhibit **good behaviors** within a configurable **time frame**
- **Strengthen traditional objectives** (correct behaviors at the limit)

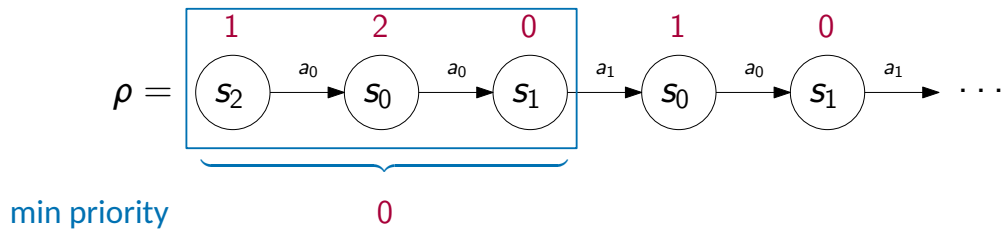
⇝ Make use of the **window formalism** to reason about **behaviors** in a given time bound $\lambda > 0$.

$$\mathrm{GW}(\lambda) = \{\rho \in \mathrm{Runs}(\mathcal{M}) \mid \textit{good behavior in at most } \lambda \textit{ steps from } s_0\}$$

Window Parity



- Minimum priority is even in at most $\lambda = 3$ steps?
- Window of maximal size $\lambda = 3$



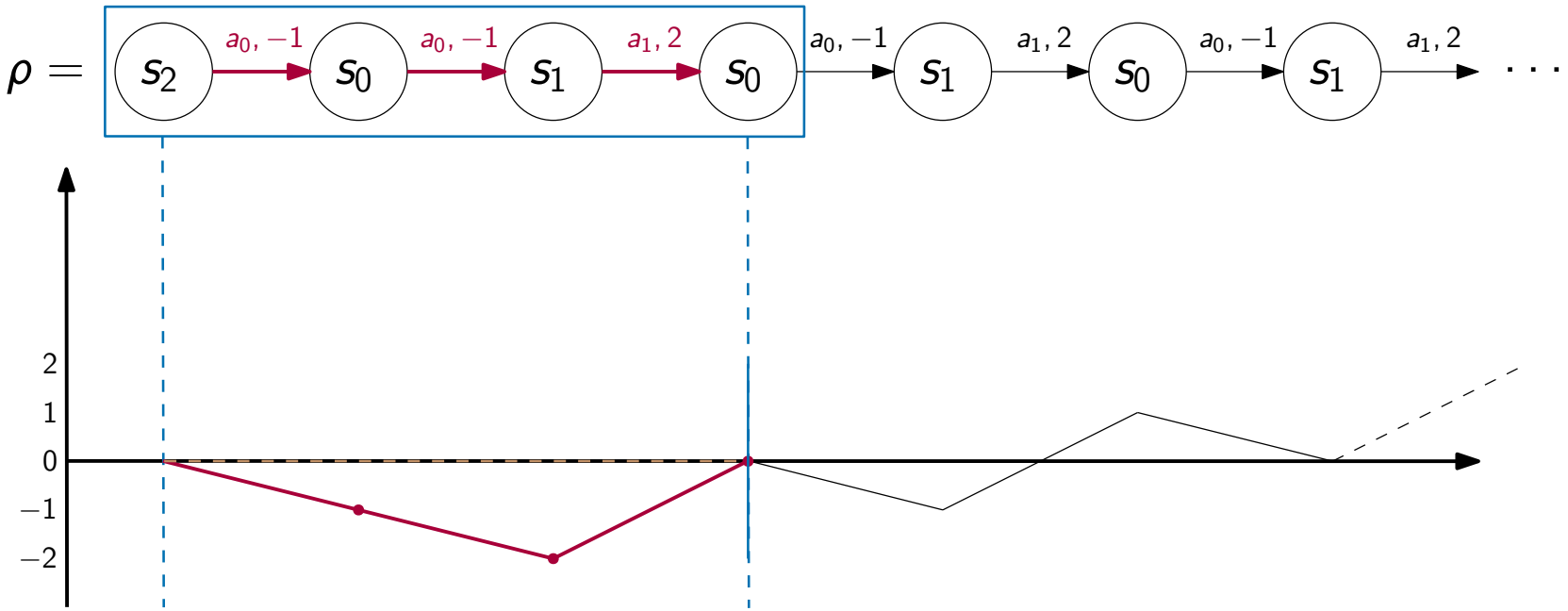
⇝ Good window of size $\lambda = 3$

⇝ $\rho \in \mathrm{GW}_{\mathrm{par}}(3)$

Window Objectives: Direct Fixed Window

- Fix a window size $\lambda > 0$
- **Direct Fixed Window objective:** $\mathrm{DFW}(\lambda) \equiv \Box \mathrm{GW}(\lambda)$
- **Good Window** of maximal size λ **sliding along the run**

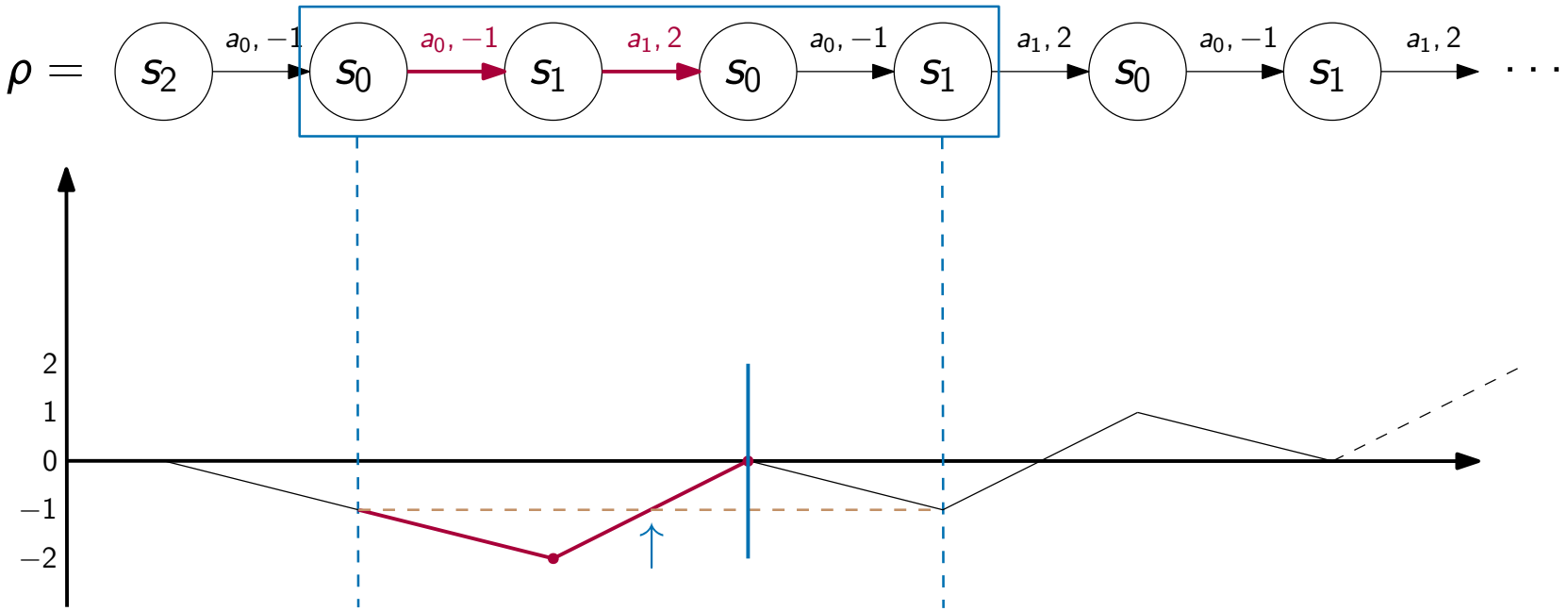
$$\rho = s_2 \xrightarrow{a_0} (s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1})^\omega \in \mathrm{DFW}(3)$$



Window Objectives: Direct Fixed Window

- Fix a window size $\lambda > 0$
- **Direct Fixed Window objective:** $\mathrm{DFW}(\lambda) \equiv \Box \mathrm{GW}(\lambda)$
- **Good Window** of maximal size λ **sliding along the run**

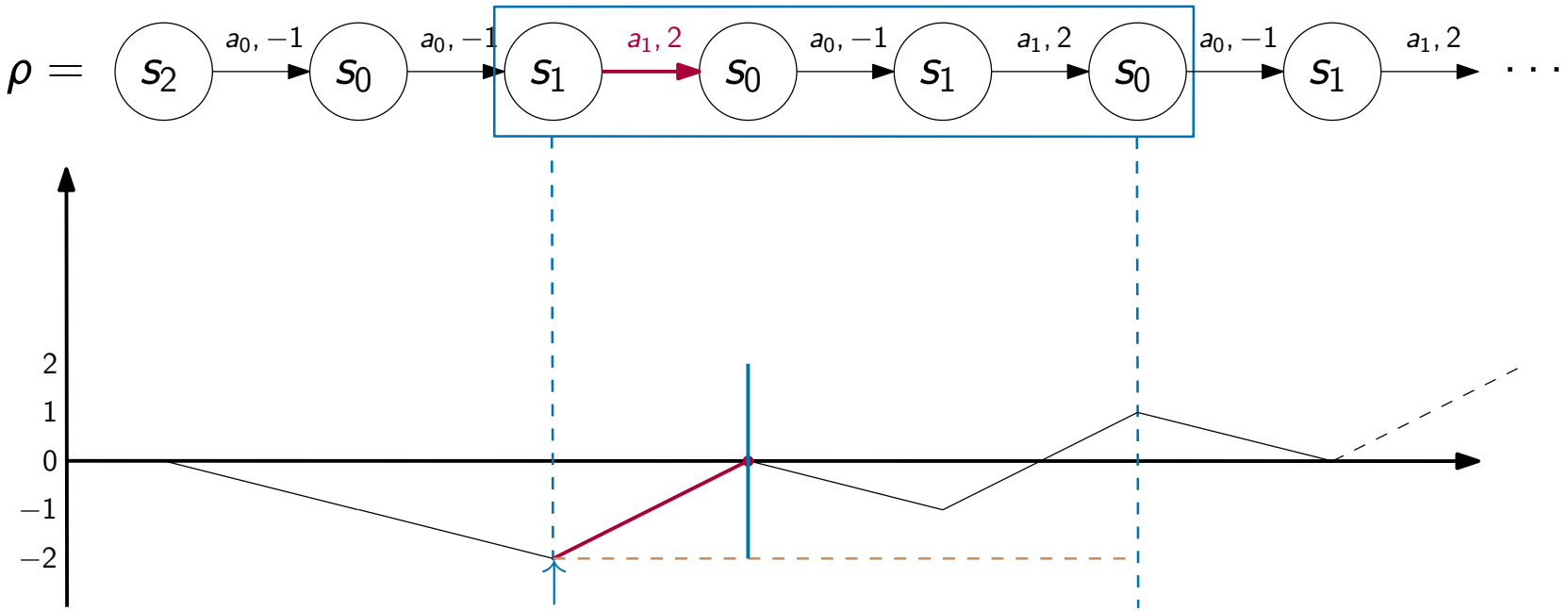
$$\rho = s_2 \xrightarrow{a_0} (s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1})^\omega \in \mathrm{DFW}(3)$$



Window Objectives: Direct Fixed Window

- Fix a window size $\lambda > 0$
- **Direct Fixed Window objective:** $\mathrm{DFW}(\lambda) \equiv \Box\mathrm{GW}(\lambda)$
- **Good Window** of maximal size λ **sliding along the run**

$$\rho = s_2 \xrightarrow{a_0} (s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1})^\omega \in \mathrm{DFW}(3)$$

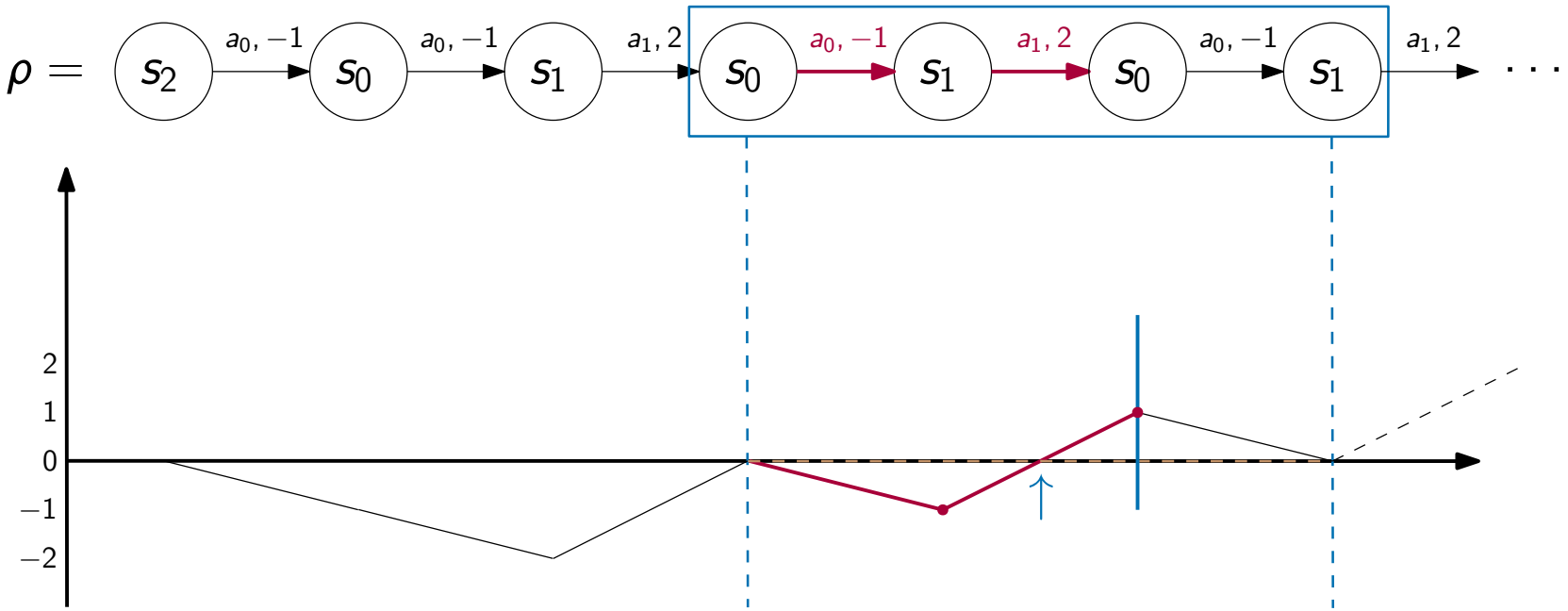


⇝ **Inductive property**

Window Objectives: Direct Fixed Window

- Fix a window size $\lambda > 0$
- **Direct Fixed Window objective:** $\mathrm{DFW}(\lambda) \equiv \Box\mathrm{GW}(\lambda)$
- **Good Window** of maximal size λ **sliding along the run**

$$\rho = s_2 \xrightarrow{a_0} (s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1})^\omega \in \mathrm{DFW}(3)$$

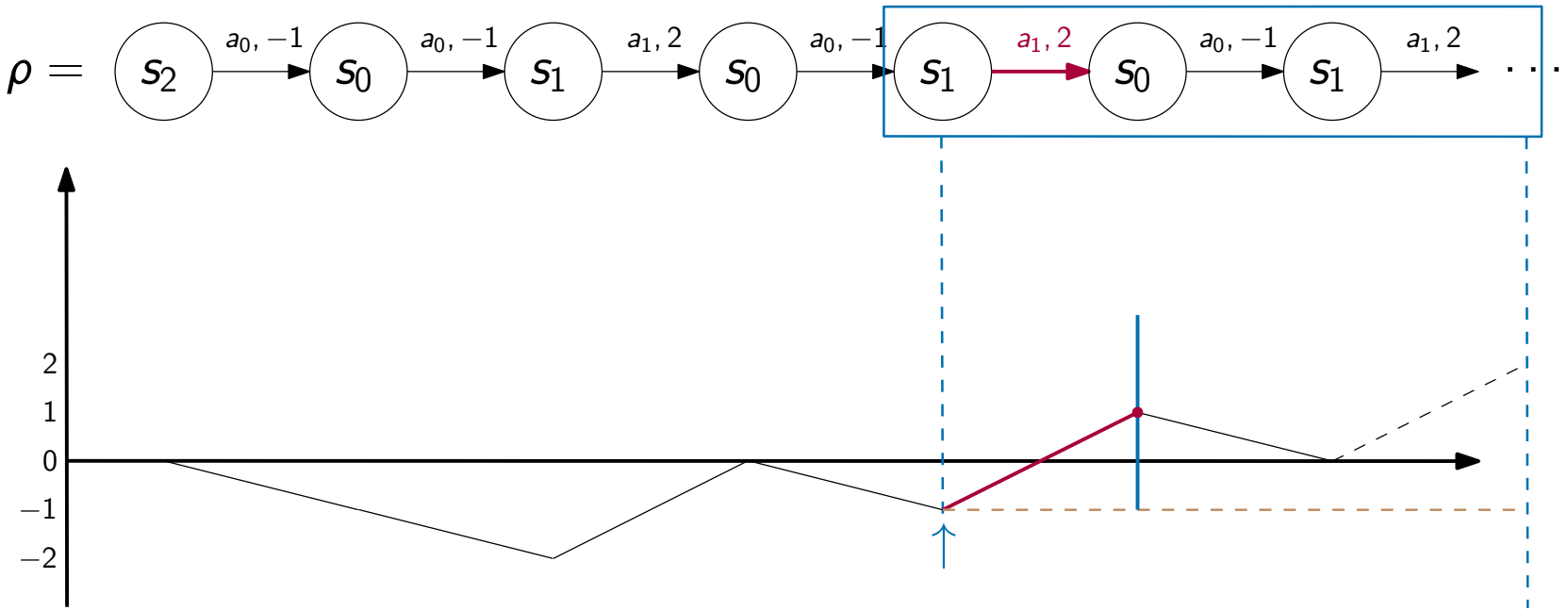


⇝ **Inductive property**

Window Objectives: Direct Fixed Window

- Fix a window size $\lambda > 0$
- **Direct Fixed Window objective:** $\mathrm{DFW}(\lambda) \equiv \Box\mathrm{GW}(\lambda)$
- **Good Window** of maximal size λ **sliding along the run**

$$\rho = s_2 \xrightarrow{a_0} (s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1})^\omega \in \mathrm{DFW}(3)$$

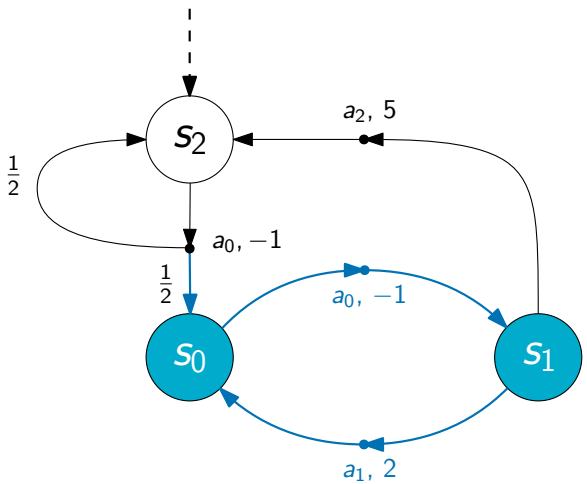


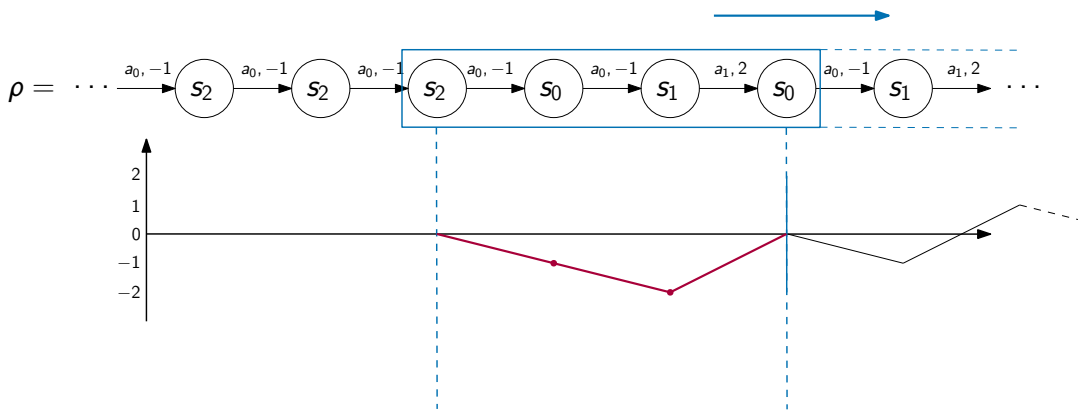
⇝ **Inductive property**

Window Objectives: prefix independence

- Window objectives at the limit
- **Fixed Window objective:** $\mathrm{FW}(\lambda) \equiv \Diamond \mathrm{DFW}(\lambda) \equiv \Diamond\Box \mathrm{GW}(\lambda)$
- **Bounded Window objective:** $\mathrm{BW} \equiv \exists \lambda > 0,\ \mathrm{FW}(\lambda)$

$$\rho = (s_2 \xrightarrow{a_0})^+ (s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1})^\omega \in \mathrm{FW}(3) \cap \mathrm{BW}$$





Threshold Probability Problem

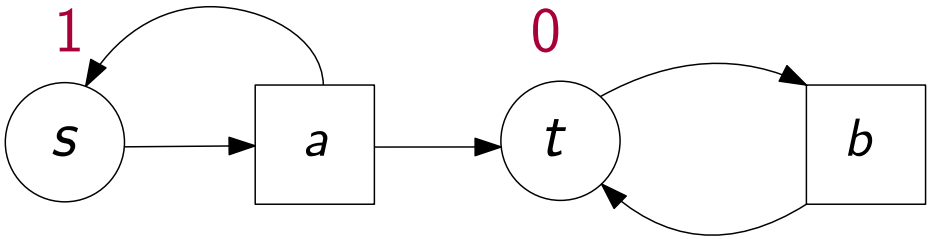
Given

- an ***MDP*** $\mathcal{M}$ with state space S,
- a ***maximal window size*** $\lambda > 0$,
- an initial ***state*** $s \in S$,
- a ***window objective*** $\mathbb{O} \in \{\mathrm{DFW}(\lambda), \mathrm{FW}(\lambda), \mathrm{BW}\}$ for mean-payoff or parity and
- a ***probability threshold*** $\alpha \in [0, 1] \cap \mathbb{Q}$,

decide if

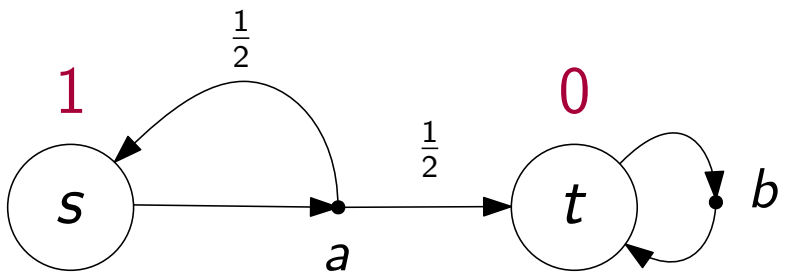
$$\exists ?\sigma \quad \mathbb{P}^{\sigma}_{\mathcal{M},s}[\mathbb{O}] \geq \alpha$$

Window Games



$\exists ? \sigma_{\bigcirc} \forall \sigma_{\square},\ \ \rho = \mathrm{Outcome}(s, \sigma_{\bigcirc}, \sigma_{\square}) \in \mathrm{DFW}(\lambda)$

- **Existence of a uniform bound $\lambda^\star$ on the maximal window size**
- $\bigcirc$ is loosing for all λ since $\square$ can choose $a \to s$



$\exists ? \sigma, \mathbb{P}_s^{\sigma}[\mathrm{DFW}(\lambda)] \geq \alpha$

- no uniform bound on the maximal window size
- For $\lambda > 1$, $\mathbb{P}_s[\mathrm{DFW}(\lambda)] = 1 - \frac{1}{2^{\lambda-1}}$
- $\mathbb{P}_s[\mathrm{DFW}(3)] = \Delta(s,a)(t) + \Delta(s,a)(s') \cdot \Delta(s,a)(t) = \frac{1}{2} + \frac{1}{4} = \frac{3}{4}$

Results overview

Markov Decision Processes

	parity		mean-payoff	
	complexity	memory	complexity	memory
DFW	P-c.	polynomial	EXPTIME/PSPACE-h.	pseudo-polynomial
FW			P-c.	polynomial
BW		memoryless	NP ∩ coNP	memoryless

Games [CDRR15, BHR16]

	parity		mean-payoff	
	complexity	memory ($\mathcal{P}_1$)	complexity	memory ($\mathcal{P}_1$)
DFW	P-c.	polynomial	P-c.	polynomial
FW				
BW		memoryless	NP ∩ coNP	memoryless

Outline

1. Model
2. Objectives
   - 2.1 Classical long-run objectives
   - 2.3 Window objectives
   - 2.5 Decision problem
3. Fixed case
   - 3.1 Reductions
   - 3.3 Direct Fixed Window
4. Prefix independent objectives
   - 4.1 The case of end-components
   - 4.3 Fixed and Bounded window
5. Extensions

Strategies: memory requirements

- ***Pure finite memory strategies*** are sufficient for the threshold probability problem
- **Main tools:** natural reduction from
  - DFW to **saftey**
  - FW to **co-Büchi**

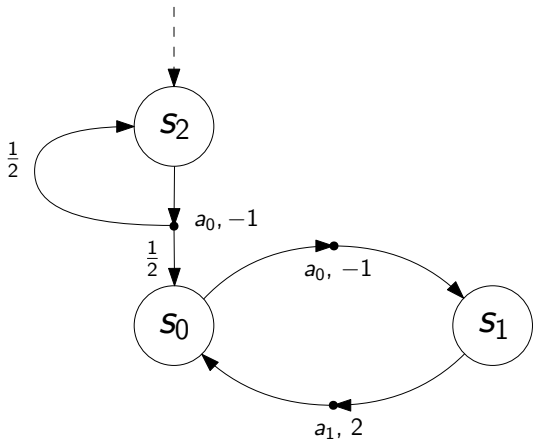
⇝ **unfolding** based on the maximal window size λ

- **Idea: incorporate weights (resp. priorities)** as well as the **current number of steps in the state space** of the MDP

Mean-Payoff	**Parity**
$S \times \{0, 1, \ldots, \lambda\} \times \{-\lambda \cdot W, \ldots, 0\}$	$S \times \{0, 1, \ldots, \lambda\} \times \{0, 1, \ldots, d\}$

Unfolding: example

MDP $\mathcal{M}$



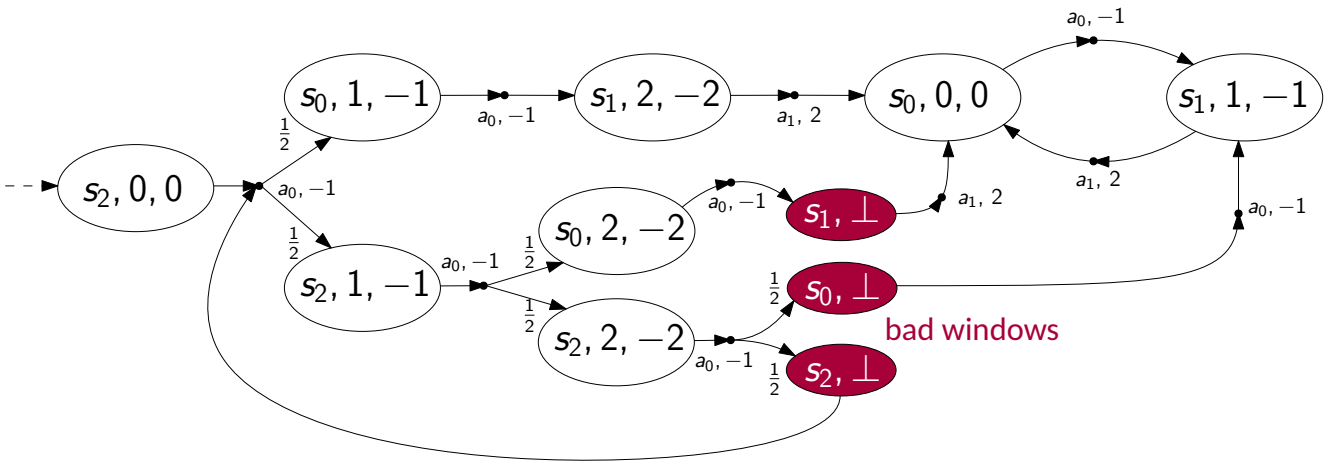
$\forall \sigma$ in $\mathcal{M}$, $\exists \tilde{\sigma}$ in $\mathcal{M}_\lambda$

$$\mathbb{P}^{\tilde{\sigma}}_{\mathcal{M}_\lambda}[\Box\neg s_\perp] = \mathbb{P}^{\sigma}_{\mathcal{M}}[\mathrm{DFW}(\lambda)] \wedge$$
$$\mathbb{P}^{\tilde{\sigma}}_{\mathcal{M}_\lambda}[\Diamond\Box\neg s_\perp] = \mathbb{P}^{\sigma}_{\mathcal{M}}[\mathrm{FW}(\lambda)]$$

$\forall \tilde{\sigma}$ in $\mathcal{M}_\lambda$, $\exists \sigma$ in $\mathcal{M}$

$$\mathbb{P}^{\sigma}_{\mathcal{M}}[\mathrm{DFW}(\lambda)] = \mathbb{P}^{\tilde{\sigma}}_{\mathcal{M}_\lambda}[\Box\neg s_\perp] \wedge$$
$$\mathbb{P}^{\sigma}_{\mathcal{M}}[\mathrm{FW}(\lambda)] = \mathbb{P}^{\tilde{\sigma}}_{\mathcal{M}_\lambda}[\Diamond\Box\neg s_\perp]$$

$\mathcal{M}_\lambda$, unfolding of $\mathcal{M}$ up to $\lambda = 3$



Complexity and memory requirements

DFW

Mean-Payoff	Parity
Threshold probability problem	
• EXPTIME algorithm	• P algorithm
• Pseudo-polynomial-memory optimal strategies	• Polynomial-memory optimal strategies
• PSPACE-hard [HK15]	• P-hard [Bee80, Imm81]
• Pseudo-polynomial-memory strategies necessary	• Polynomial-memory strategies necessary

Outline

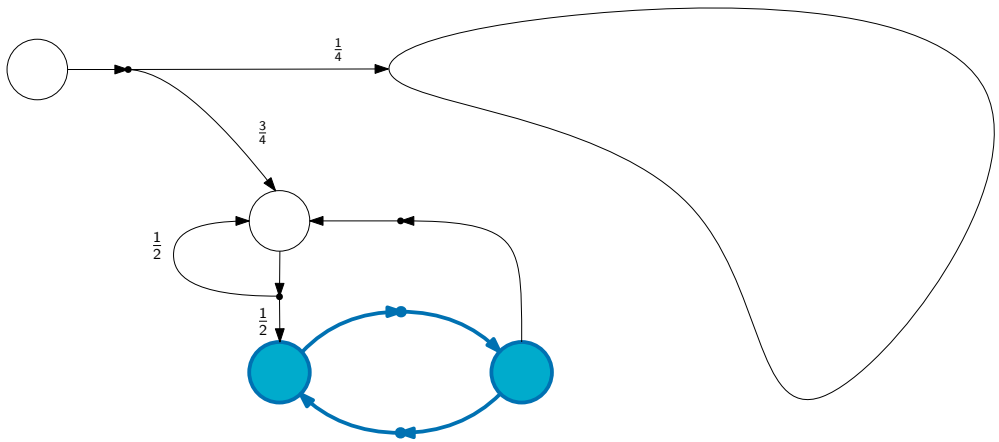
1. Model
2. Objectives
   - 2.1 Classical long-run objectives
   - 2.3 Window objectives
   - 2.5 Decision problem
3. Fixed case
   - 3.1 Reductions
   - 3.3 Direct Fixed Window
4. Prefix independent objectives
   - 4.1 The case of end-components
   - 4.3 Fixed and Bounded window
5. Extensions

End-component

Let $\mathcal{M}$ be an MDP, an ***end-component (EC)*** is a **strongly connected** sub-MDP $\mathcal{C}$ of $\mathcal{M}$ formed by states and actions allowing to never leave $\mathcal{C}$

- For any strategy σ, all runs ρ compatible with σ end up in an EC with probability one

$\rightsquigarrow$ the set of states and actions seen infinitely often in ρ form an EC with probability one



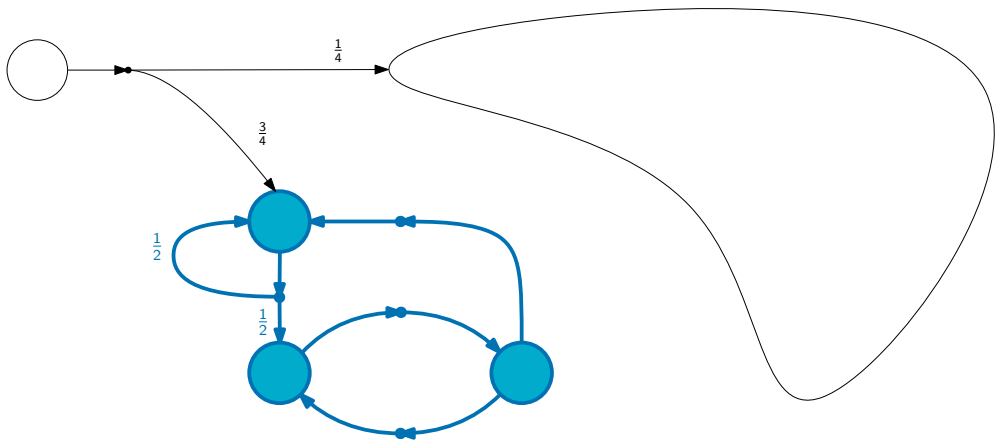
End-component

Let $\mathcal{M}$ be an MDP, an ***end-component (EC)*** is a **strongly connected** sub-MDP $\mathcal{C}$ of $\mathcal{M}$ formed by states and actions allowing to never leave $\mathcal{C}$

- The number of ECs may be exponential in the size of $\mathcal{M}$
- An EC may have sub-ECs
- the union of two ECs with non-empty intersection is an EC

$\Longrightarrow$ **Maximal end-component (MEC)** = ECs that cannot be extended

$\rightsquigarrow$ MEC($\mathcal{M}$) computable in polynomial time



MEC classification and Zero-one Law

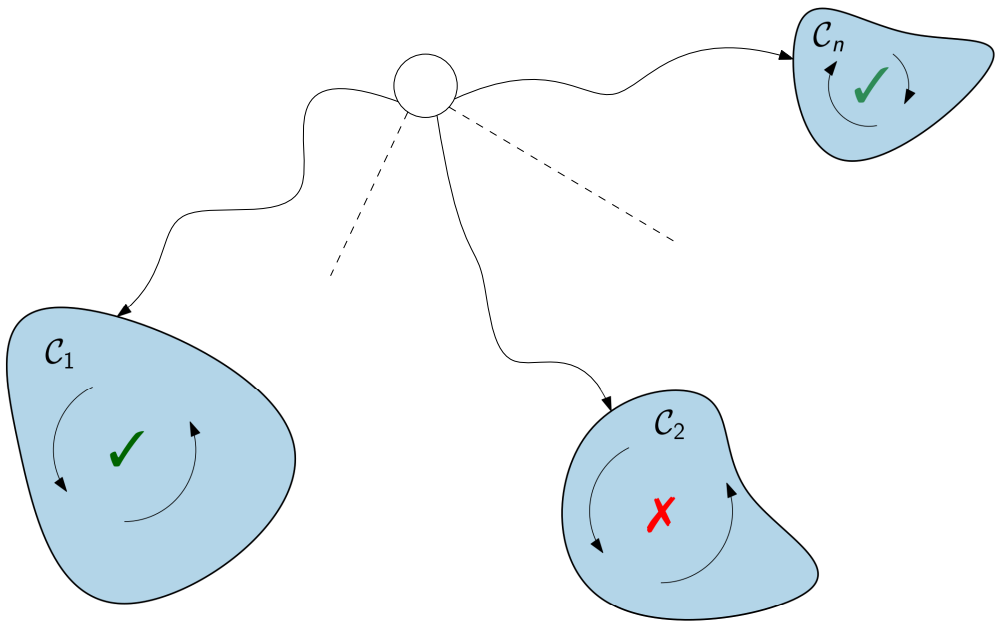
- **Prefix independence** $\rightsquigarrow$ MECs analysis
- **Main result**: ***MEC classification***
- **strong link between ECs and 2 player games**

$\rightsquigarrow$ 2 types of MECs: ✓ and ✗

Given an objective $\mathbb{O} \in \{\mathrm{FW}(\lambda), \mathrm{BW}\}$

✓ $\forall s$ of $\mathcal{C}$ $\exists \sigma$, $\quad \mathbb{P}^{\sigma}_{\mathcal{C},s}[\mathbb{O}] = 1$

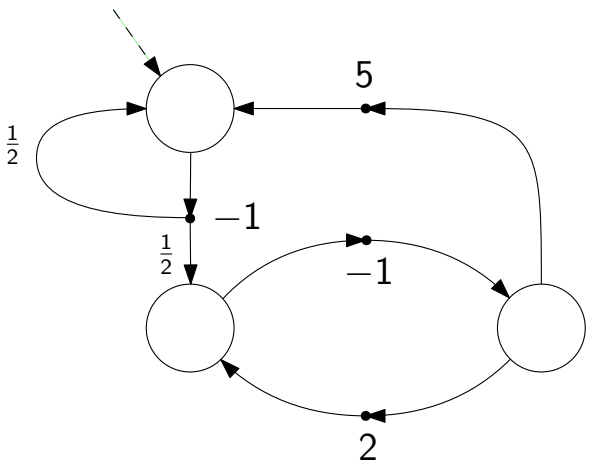
✗ $\forall s$ of $\mathcal{C}$ $\forall \sigma$, $\quad \mathbb{P}^{\sigma}_{\mathcal{C},s}[\mathbb{O}] = 0$

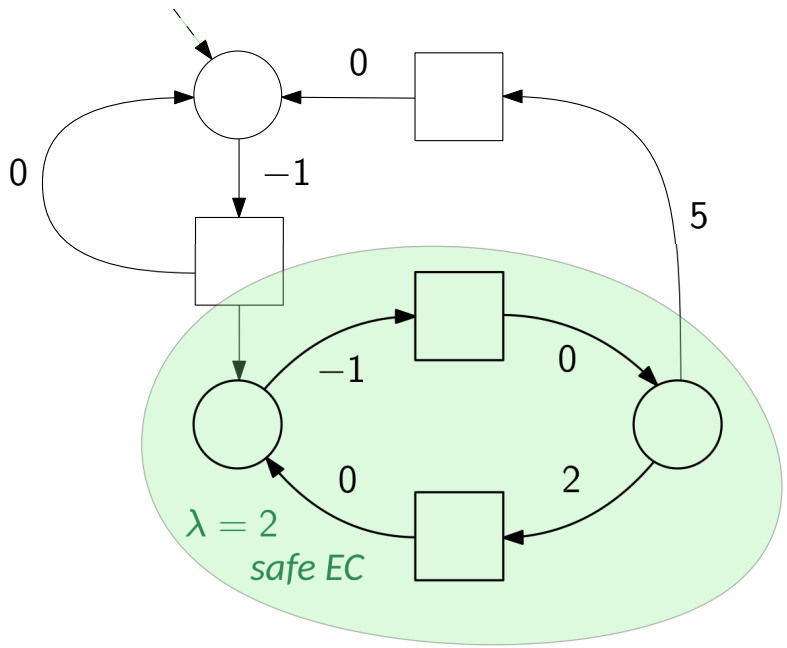


Safe EC

An EC $\mathcal{C}$ of state space $S_{\mathcal{C}}$ is λ-*safe* iff $\forall s \in S_{\mathcal{C}}$,
$\exists \sigma\ \forall \rho \in \mathrm{Runs}^{\sigma}(\mathcal{C}),\ \rho \in \mathrm{DFW}(\lambda)$

- ⇝ Boils down to interpreting $\mathcal{C}$ as a 2 player game
- Determine if $\mathcal{C}$ is safe ?
  - → compute the winning set $\mathcal{W}_{\mathrm{dfw}}$ of the DFW game version of $\mathcal{C}$
  - $\mathcal{W}_{\mathrm{dfw}} = S_{\mathcal{C}} \implies \mathcal{C}$ is safe

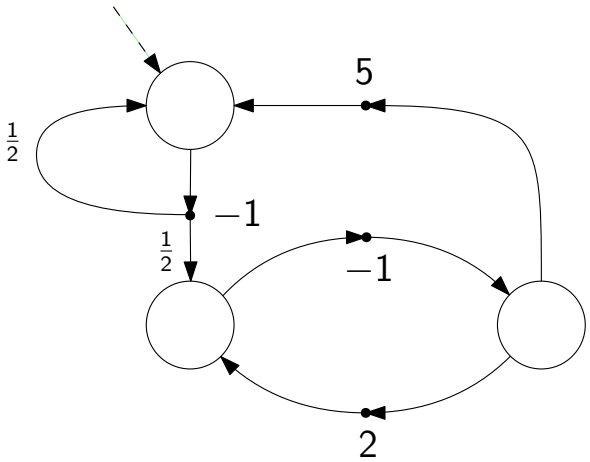


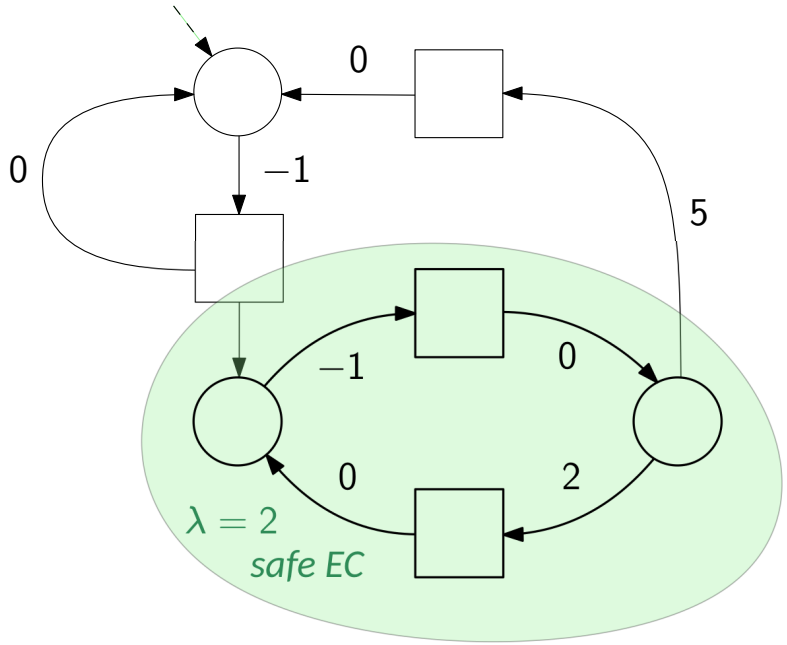


Safe EC

An EC $\mathcal{C}$ of state space $S_{\mathcal{C}}$ is λ-*safe* iff $\forall s \in S_{\mathcal{C}}$,
$\exists \sigma\ \forall \rho \in \mathrm{Runs}^{\sigma}(\mathcal{C}),\ \rho \in \mathrm{DFW}(\lambda)$

- ⇝ Boils down to interpreting $\mathcal{C}$ as a 2 player game
- $\exists ?\mathcal{C}$, λ-safe EC inside a super-EC $\mathcal{C}^{\star}$
  - → compute the winning set $\mathcal{W}_{\mathrm{dfw}}$ of the DFW game version of $\mathcal{C}^{\star}$
    - $\mathcal{W}_{\mathrm{dfw}} \neq \emptyset \implies \exists\ \lambda$-safe EC $\mathcal{C}$ inside $\mathcal{C}^{\star}$

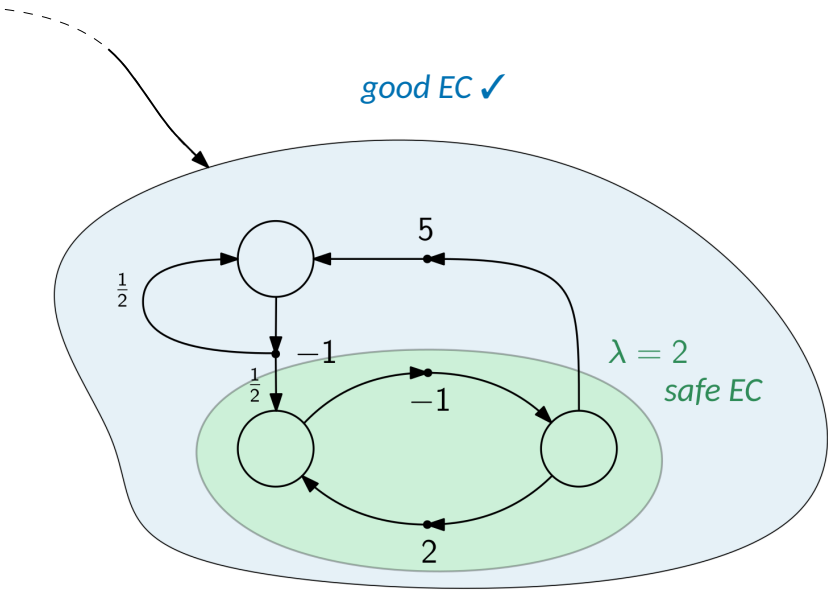




Good EC

An EC $\mathcal{C}$ is

- ✓ λ-good for $\lambda > 0$ if it contains a sub-EC $\mathcal{C}'$ which is λ-safe
- ✓ BW-good if it contains a sub-EC $\mathcal{C}'$ which is λ-safe for some $\lambda > 0$



- In all EC $\mathcal{C}$, $\exists \sigma^{\mathcal{C}}_{\text{visit}}$ allowing to visit with probability one all states of $\mathcal{C}$

$\Longrightarrow$ $\mathbb{P}^{\sigma^{\mathcal{C}}_{\text{visit}}}_{\mathcal{C}}(\Diamond \mathcal{C}_{safe}) = 1$

- $\sigma_{\checkmark}$ = combine $\sigma^{\mathcal{C}}_{\text{visit}}$ and σ_{safe}

$\Longrightarrow$ $\mathbb{P}^{\sigma_{\checkmark}}_{\mathcal{C}}[\mathbb{O}] = 1$ for $\mathbb{O} \in \{\text{FW}(\lambda), \text{BW}\}$

BW-goodness

- **Uniform bound $\lambda^\star$ in window games**: window size $\lambda^\star$ is sufficiently large so that the bounded version coincides with the fixed one
- Win DFW for $\lambda^\star$ inside $\mathcal{C} \implies \mathcal{C}$ is BW-good ✓
- $\lambda^\star$ polynomial for **window parity games**
- $\lambda^\star$ pseudo-polynomial for **window mean-payoff** games

Can we do better ?

BW-goodness

- **Uniform bound $\lambda^\star$ in window games**: window size $\lambda^\star$ is sufficiently large so that the bounded version coincides with the fixed one
- Win DFW for $\lambda^\star$ inside $\mathcal{C} \implies \mathcal{C}$ is BW-good ✓
- $\lambda^\star$ polynomial for **window parity games**
- $\lambda^\star$ pseudo-polynomial for **window mean-payoff** games

Can we do better ? **Yes!** [CHH09, CDRR15, BHR16]

Bad Window =

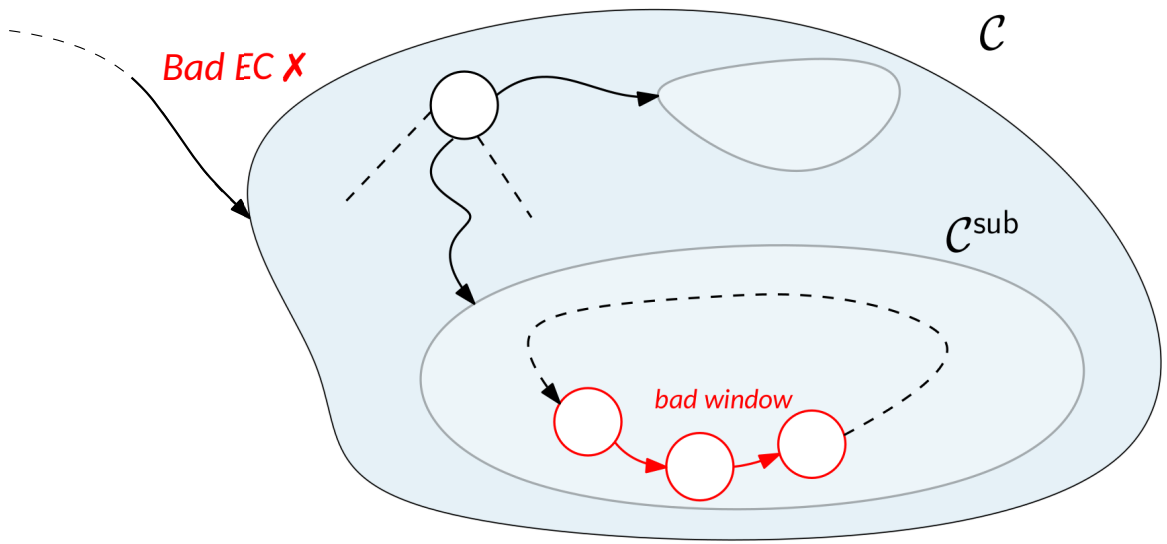
Mean-Payoff $\neg$Total Payoff $= \left\{\rho = s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} \cdots \mid \limsup_{n\to\infty} \sum_{i=0}^{n} w(a_i) < 0\right\}$

Parity co-Weak Parity $= \left\{\rho = s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} \cdots \mid \min_{i>0}[p(s_i)] \text{ is odd}\right\}$

- $\exists \lambda > 0,\ \mathrm{DFW}(\lambda) \equiv \exists \lambda > 0,\ \Box\mathrm{GW}(\lambda) \equiv \neg\Diamond$Bad Window
- **Classification**: compute the **safe region** $\mathcal{W}_{\text{safe}}$ w.r.t. Bad Window of the 2PG version of $\mathcal{C}$
  - $\mathcal{W}_{\text{safe}} \neq \emptyset \implies \mathcal{C}$ is BW-good ✓ with σ_{safe}, a memoryless (safety) strategy

Bad EC ?

- $\nexists$ safe sub-EC inside the 2PG version of $\mathcal{C}$?
- Fix any finite-memory strategy σ inside $\mathcal{C}$
- The set of states and actions seen infinitely often form sub-ECs $\mathcal{C}^{\text{sub}}$ with probability one
- In these sub-ECs $\mathcal{C}^{\text{sub}}$, $\exists \rho \notin \text{DFW}(\lambda)$ (otherwise $\mathcal{C}^{\text{sub}}$ is safe)
- extract a bad prefix $\rho^{\text{✗}}$ (=bad window) in ρ
- $\rho^{\text{✗}}$ is repeated infinitely often with probability one in $\mathcal{C}^{\text{sub}}$
- $\forall \mathcal{C}^{\text{sub}},\ \mathbb{P}^{\sigma}_{\mathcal{C}^{\text{sub}}}[\text{DFW}(\lambda)] = 0 \implies \mathbb{P}^{\sigma}_{\mathcal{C}}[\mathbb{O}] = 0 \qquad \mathbb{O} \in \{\text{FW}(\lambda), \text{BW}\}$



Summary: complexity and strategies

Given $\mathbb{O} \in \{\mathsf{FW}(\lambda), \mathsf{BW}\}$, **2 types of MECs: ✓ and ✗**

✓ $\forall s$ of $\mathcal{C}$ $\exists \sigma$, $\mathbb{P}^{\sigma}_{\mathcal{C},s}[\mathbb{O}] = 1$ ✗ $\forall s$ of $\mathcal{C}$ $\forall \sigma$, $\mathbb{P}^{\sigma}_{\mathcal{C},s}[\mathbb{O}] = 0$

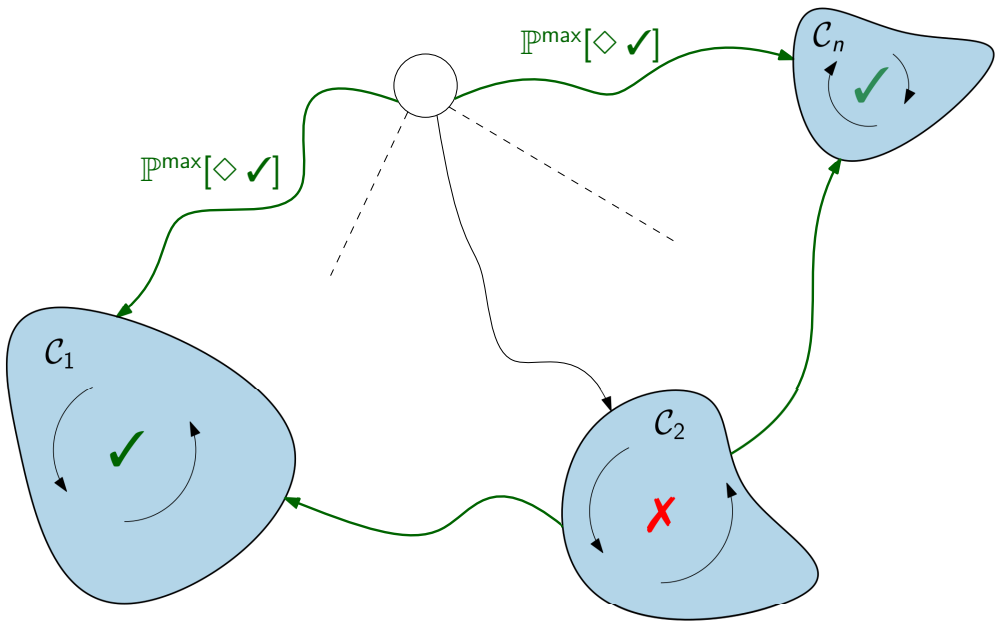
MEC classification

Mean-Payoff	Parity
Fixed window (FW)	
• in P (2PG)	• in P (2PG)
• pure polynomial finite-memory strategy $\sigma_{\checkmark} := \sigma^{\mathcal{C}}_{\text{visit}} + \sigma_{safe}$	
Bounded window (BW)	
• in NP ∩ coNP (2PG)	• in P (2PG)
• pure memoryless strategy $\sigma_{\checkmark} := \sigma^{\mathcal{C}}_{\text{visit}} + \sigma_{safe}$	

Optimal strategies for FW and BW

$$\sigma^{\max}_{\Diamond\checkmark} = \arg\max_{\sigma} \mathbb{P}^{\sigma}[\Diamond\checkmark]$$

$$\sigma^{\star} := \underbrace{\sigma^{\max}_{\Diamond\checkmark}}_{\text{until reaching } \checkmark} + \underbrace{\sigma_{\checkmark}}_{\text{in } \checkmark}$$



Summary

Markov Decision Processes

	parity		mean-payoff	
	complexity	memory	complexity	memory
DFW	P-c.	polynomial	EXPTIME/PSPACE-h.	pseudo-polynomial
FW			P-c.	polynomial
BW		memoryless	NP ∩ coNP	memoryless

Outline

1. Model
2. Objectives
   - 2.1 Classical long-run objectives
   - 2.3 Window objectives
   - 2.5 Decision problem
3. Fixed case
   - 3.1 Reductions
   - 3.3 Direct Fixed Window
4. Prefix independent objectives
   - 4.1 The case of end-components
   - 4.3 Fixed and Bounded window
5. Extensions

- ***Expected window size***: strategies maintaining the best time bounds possible in their local environment
  - $\min_\sigma \mathbb{E}^\sigma_{\mathcal{M}}[\lambda] = \min_\sigma \sum_{\lambda>0}^{\infty} \lambda \cdot \mathbb{P}^\sigma_{\mathcal{M}}[\mathrm{FW}(\lambda) \setminus \mathrm{FW}(\lambda-1)]$
  - **Refine the classification process**: identify best window size λ in each MEC by binary search
  - **Contract each good MEC and assign λ as entering weight**
- ***Multi window objectives***:
  - DFW: extends the unfolding for multiple dimensions
  - **MEC classification**: games with multiple window objectives
- ***Tool support:*** *implementation in* ***STORM***
  - \+ DFW, FW and BW objectives for parity and mean-payoff
  - \+ Efficient unfoldings for DFW
  - \+ Window games for parity [BHR16] and mean-payoff [CDRR15]
  - \+ Total payoff games: efficient pseudo-polynomial time algorithm with value iteration [BGHM14]
  - \+ Weak parity games [CHH09]
  - \+ Rich MEC classification methods
  - \+ Strategy synthesis (export in .dot format)

References I

[Bee80] Catriel Beeri, *On the membership problem for functional and multivalued dependencies in relational databases*, ACM Trans. Database Syst. **5** (1980), no. 3, 241–259.

[BGHM14] Thomas Brihaye, Gilles Geeraerts, Axel Haddad, and Benjamin Monmege, *To reach or not to reach? efficient algorithms for total-payoff games*, CoRR **abs/1407.5030** (2014).

[BHR16] Véronique Bruyère, Quentin Hautem, and Mickael Randour, *Window parity games: an alternative approach toward parity games with time bounds*, Proceedings of the Seventh International Symposium on Games, Automata, Logics and Formal Verification, GandALF 2016, Catania, Italy, 14-16 September 2016. (Domenico Cantone and Giorgio Delzanno, eds.), EPTCS, vol. 226, 2016, pp. 135–148.

[CDRR15] Krishnendu Chatterjee, Laurent Doyen, Mickael Randour, and Jean-François Raskin, *Looking at mean-payoff and total-payoff through windows*, Inf. Comput. **242** (2015), 25–52.

References II

[CHH09] Krishnendu Chatterjee, Thomas A. Henzinger, and Florian Horn, *Finitary winning in omega-regular games*, ACM Trans. Comput. Log. **11** (2009), no. 1, 1:1–1:27.

[HK15] Christoph Haase and Stefan Kiefer, *The odds of staying on budget*, Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part II (Magnús M. Halldórsson, Kazuo Iwama, Naoki Kobayashi, and Bettina Speckmann, eds.), Lecture Notes in Computer Science, vol. 9135, Springer, 2015, pp. 234–246.

[Imm81] Neil Immerman, *Number of quantifiers is better than number of tape cells*, J. Comput. Syst. Sci. **22** (1981), no. 3, 384–406.