From Iteration to System Failure

Characterizing the FITness of Periodic Weakly-Hard Systems

Arpan Gujarati*, Mitra Nasri#, Rupak Majumdar*, and Björn B. Brandenburg*

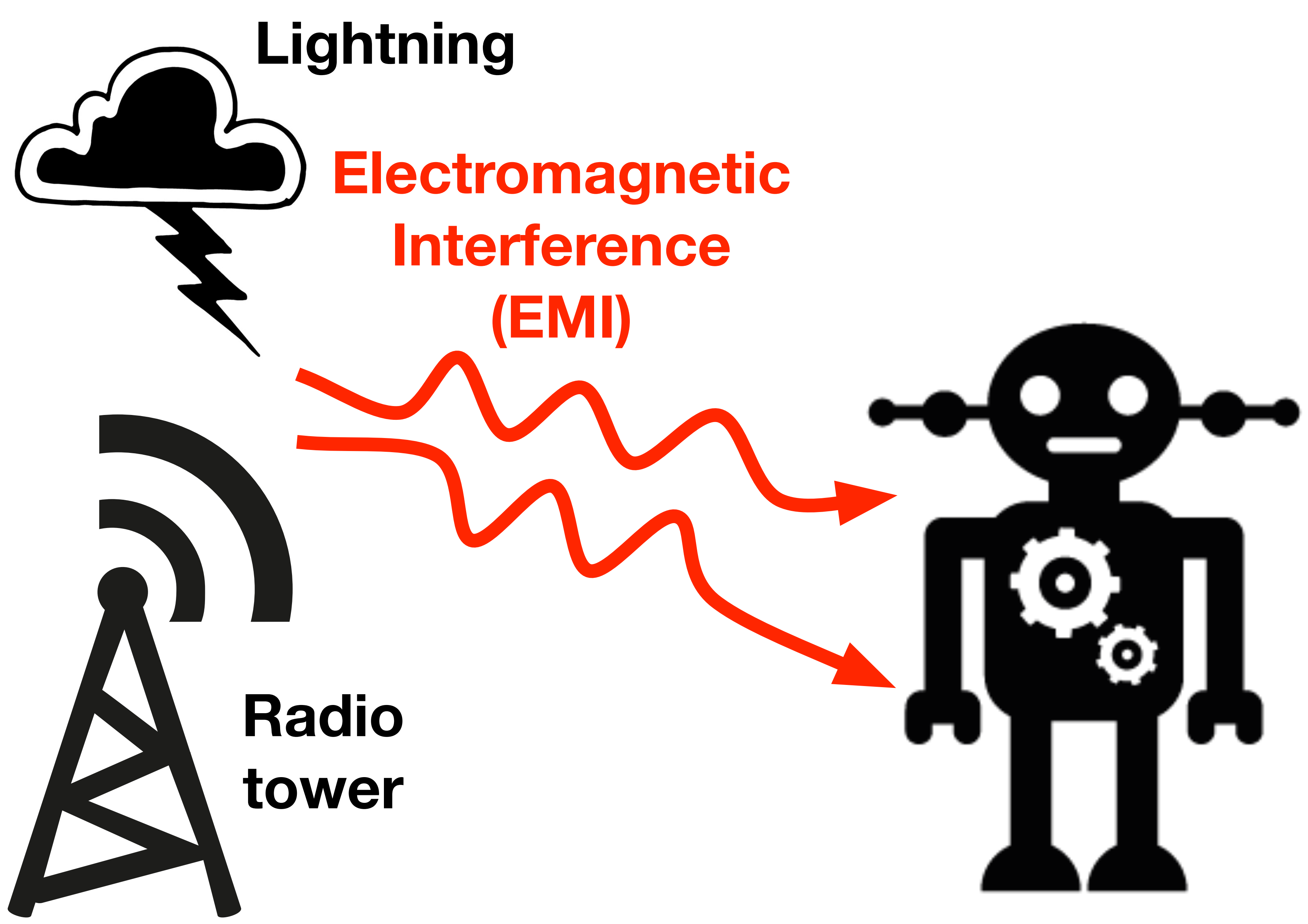
*MPI-SWS (Germany), #TU Delft (Netherlands)

MAX PLANCK INSTITUTE FOR SOFTWARE SYSTEMS

TU Delft

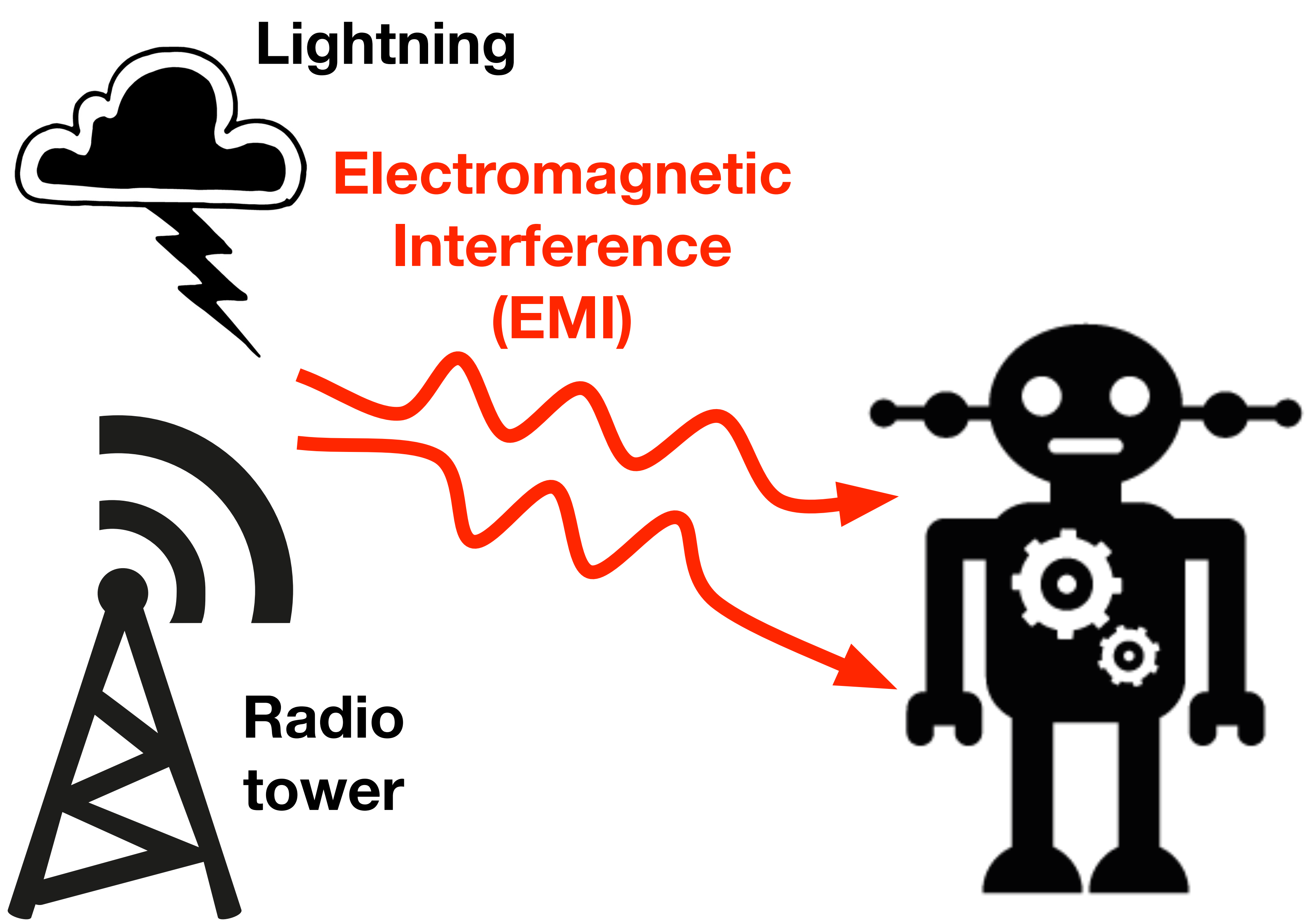
Quantitative Reliability Analysis is Essential for Safety-Critical CPS

Quantitative Reliability Analysis is Essential for Safety-Critical CPS



Zero risk of failures can never be achieved

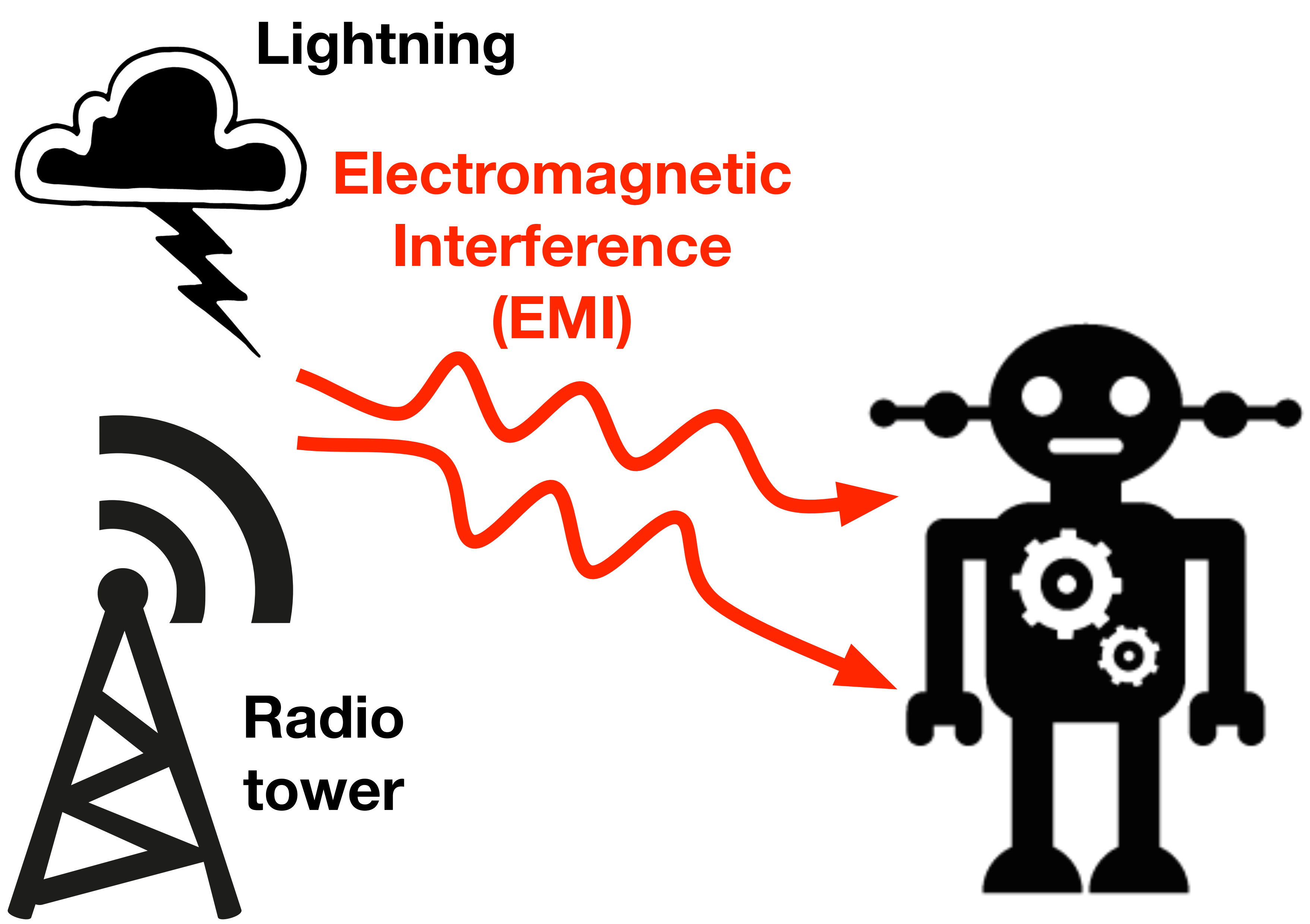
Quantitative Reliability Analysis is Essential for Safety-Critical CPS



Zero risk of failures can never be achieved

Safety certification objective: Ensure **“negligible”** failure rates

Quantitative Reliability Analysis is Essential for Safety-Critical CPS



Zero risk of failures can never be achieved

Safety certification objective:
Ensure **"negligible"** failure rates

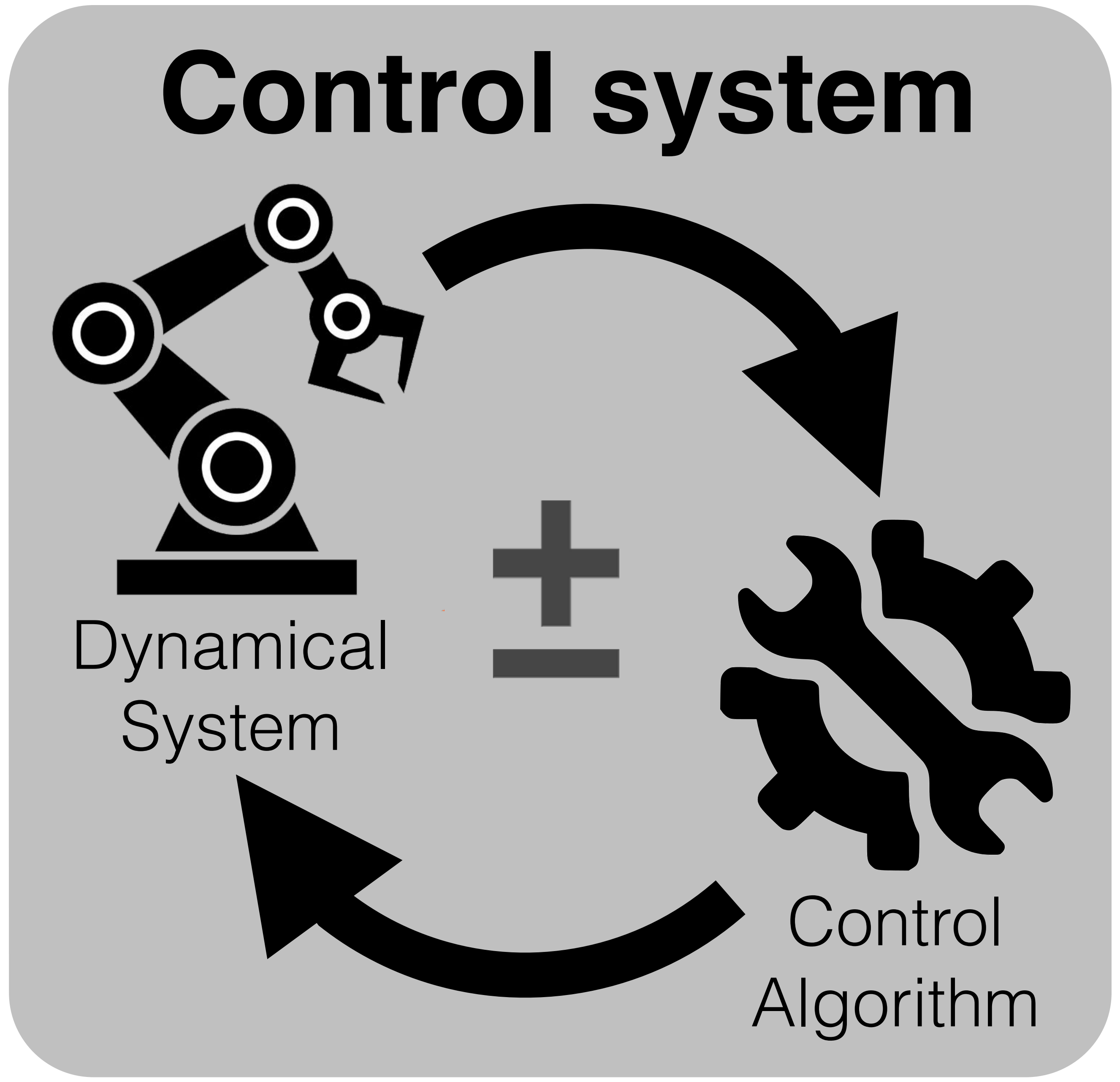
SAE INTERNATIONAL

E.g., for critical subsystems:
Pr[failure / hour] < 10^{-9}

ARP4761 and the Safety Assessment Process for Civil Airborne Systems

How to Analyse the Reliability of **Temporally Robust Systems?**

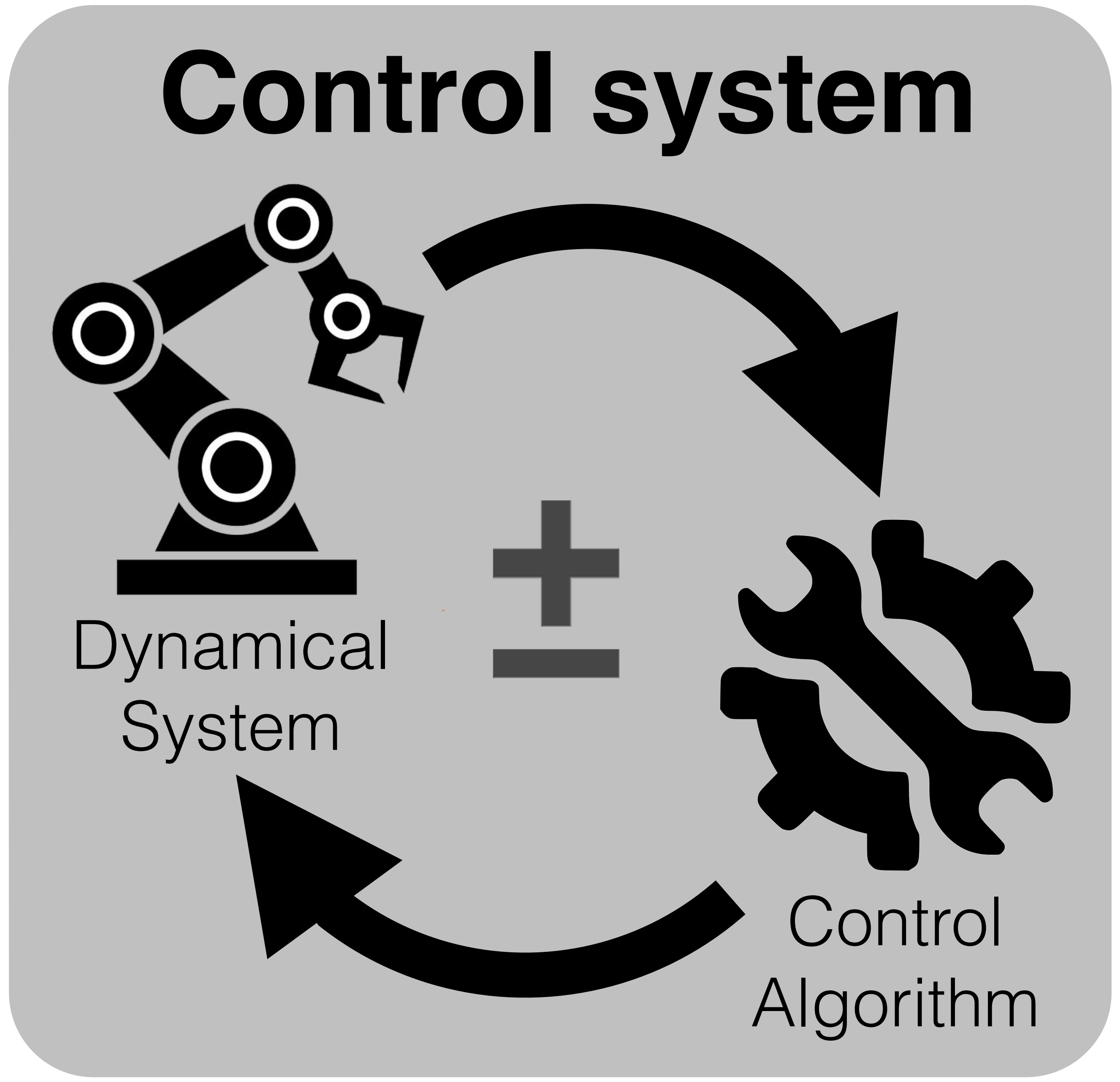
How to Analyse the Reliability of **Temporally Robust Systems?**



Motivating example

- ➡ **Frequency:** 100 Hz (10 ms time period)
- ➡ **Stability requirement:** 3 out of 4 iterations execute on time

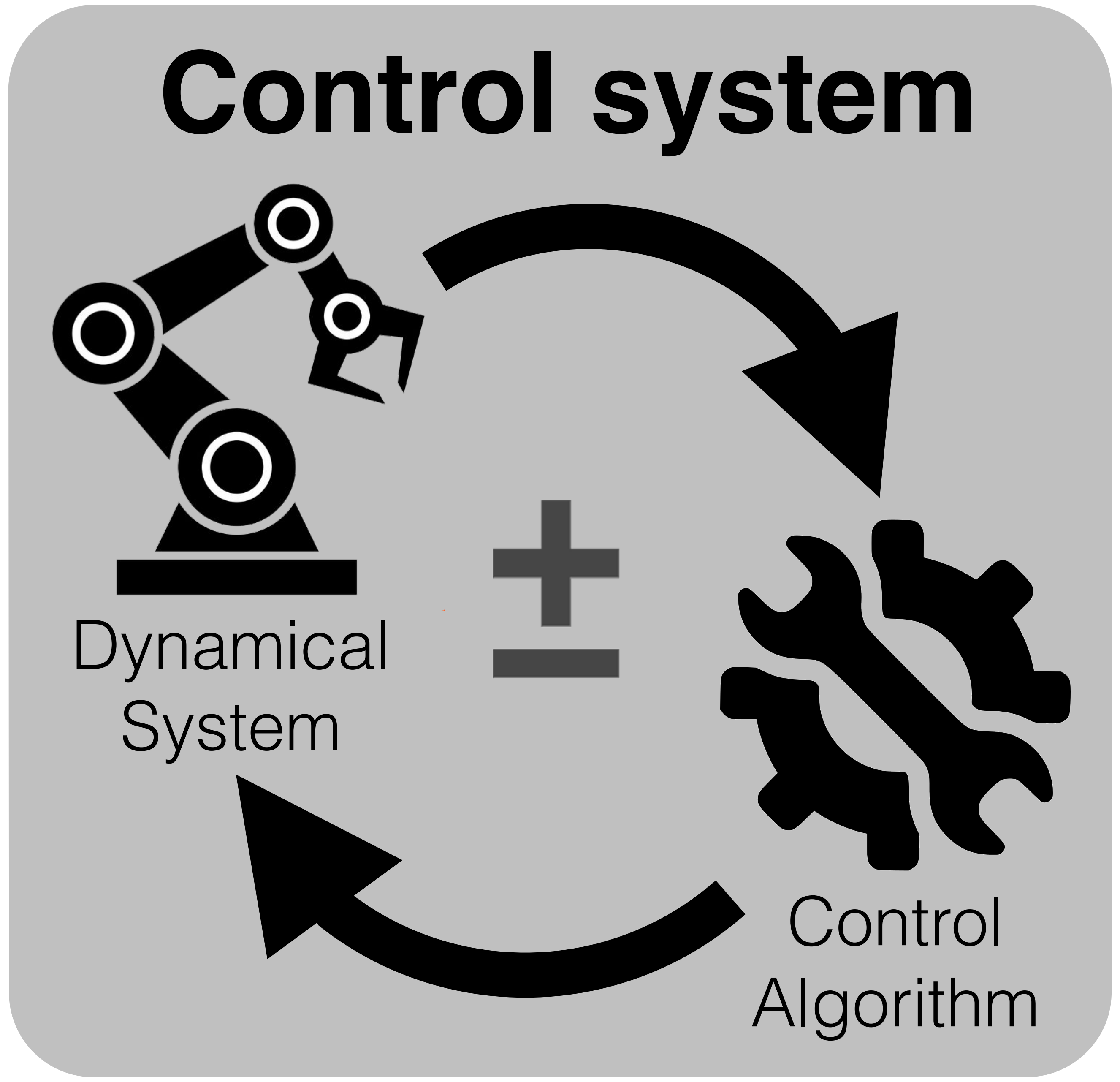
How to Analyse the Reliability of **Temporally Robust Systems?**



Motivating example

- **Frequency:** 100 Hz (10 ms time period)
- **Stability requirement:** 3 out of 4 iterations execute on time
- **Schedulability analyses:** Pr[single iteration delayed] $\leq 10^{-10}$

How to Analyse the Reliability of **Temporally Robust Systems?**



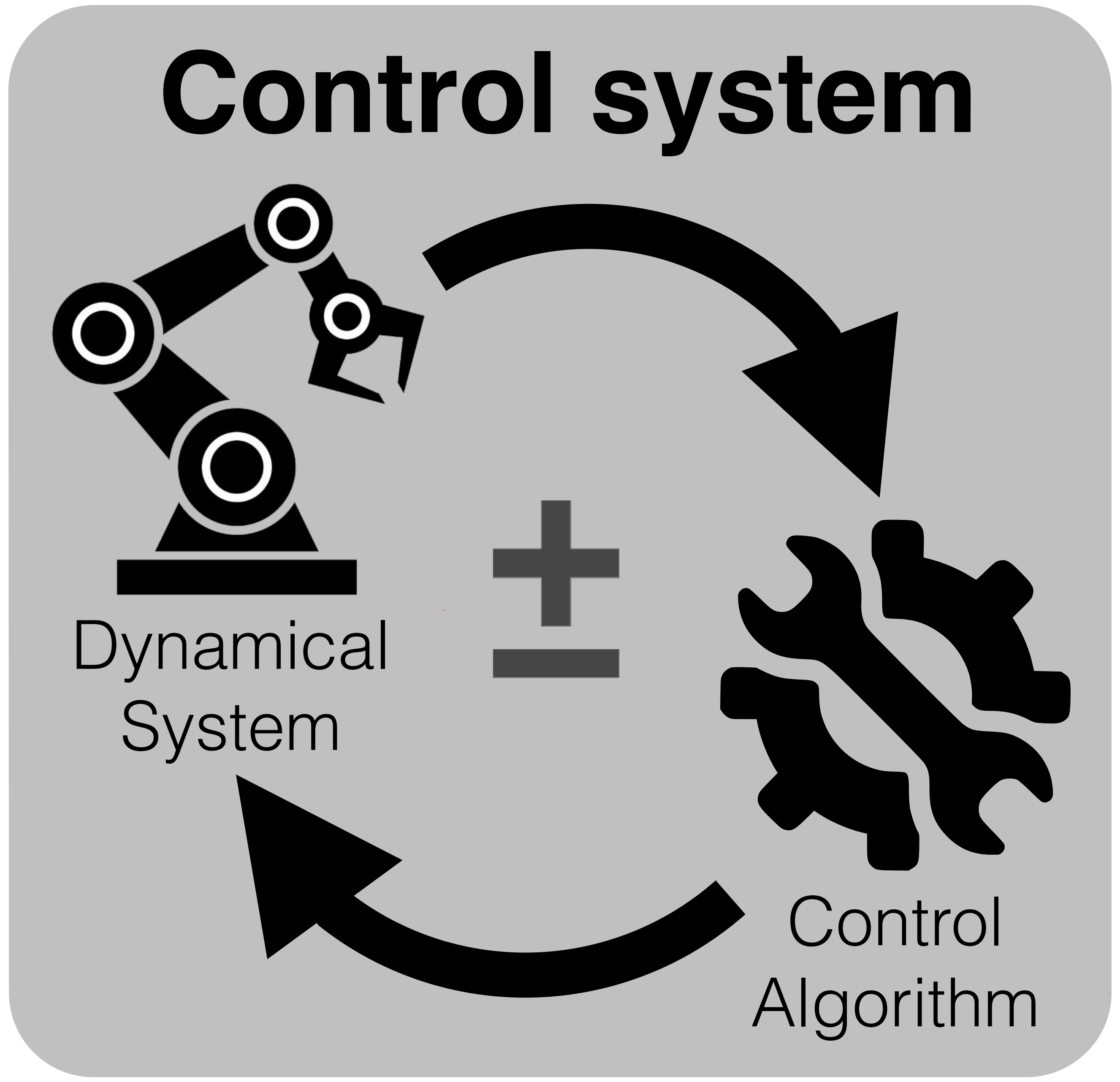
Motivating example

- **Frequency:** 100 Hz (10 ms time period)
- **Stability requirement:** 3 out of 4 iterations execute on time
- **Schedulability analyses:** Pr[single iteration delayed] ≤ 10^{-10}

Per-iteration analyses yield pessimistic failure rates

- Computing mean time to first failed iteration ignores stability requirements
- E.g., iteration failure probability of 10^{-10} ↦ **36,000 x 10^{-9} failures / hours**

How to Analyse the Reliability of **Temporally Robust Systems?**



Motivating example

- **Frequency:** 100 Hz (10 ms time period)
- **Stability requirement:** 3 out of 4 iterations execute on time
- **Schedulability analyses:** Pr[single iteration delayed] $\leq 10^{-10}$

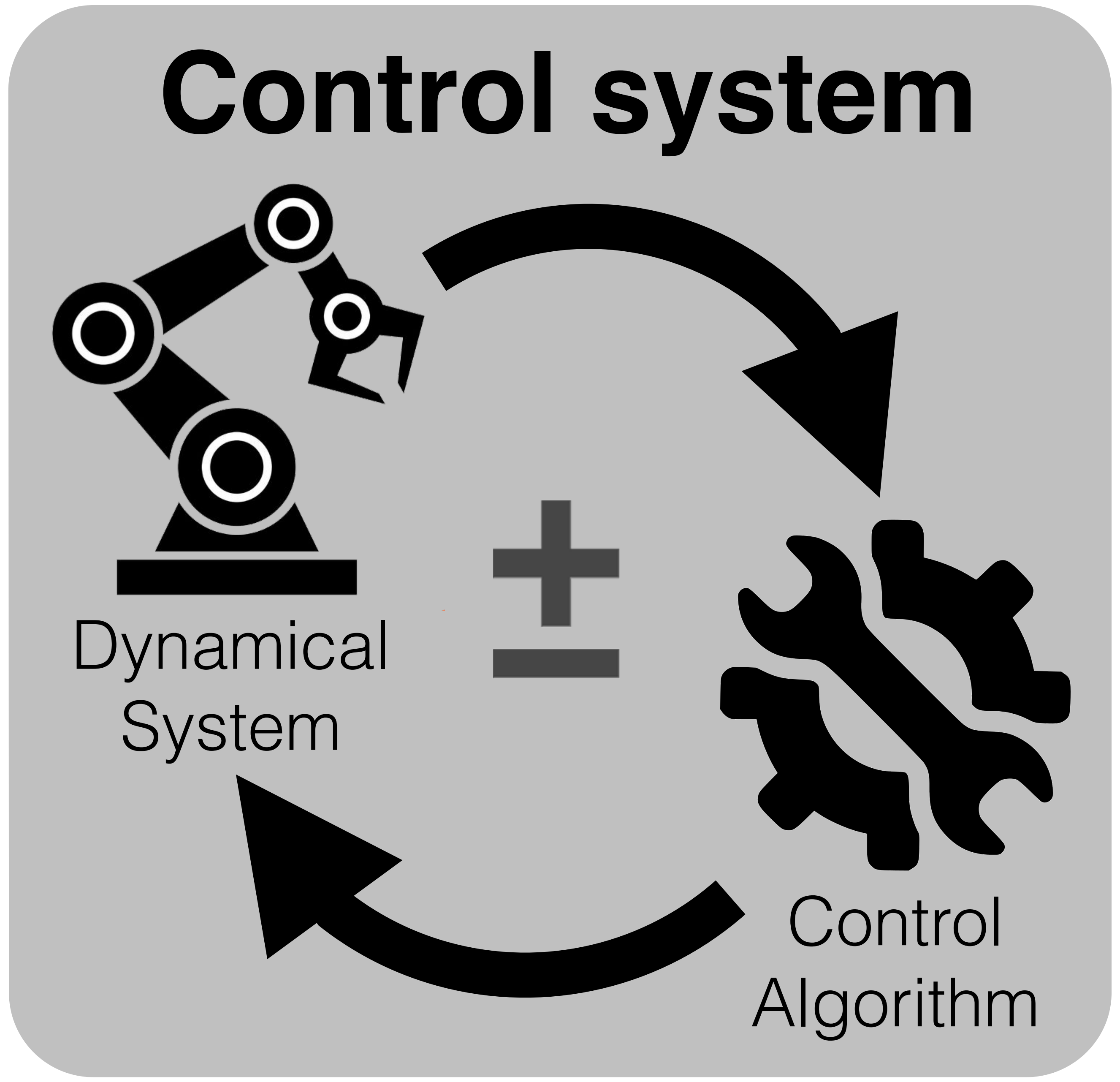
Per-iteration analyses yield pessimistic failure rates

- Computing mean time to first failed iteration ignores stability requirements
- E.g., iteration failure probability of $10^{-10} \mapsto$ **36,000 x 10^{-9} failures / hours**

Explicitly accounting for the stability requirements

- Yields more accurate failure rates
- E.g., iteration failure probability of 10^{-10} and stability requirement $\mapsto$ **1.08 x 10^{-15} failures / hours**

How to Analyse the Reliability of **Temporally Robust Systems?**



Motivating example

- ➡ **Frequency:** 100 Hz (10 ms time period)
- ➡ **Stability requirement:** 3 out of 4 iterations execute on time
- ➡ **Schedulability analyses:** Pr[single iteration delayed] ≤ 10^{-10}

Per-iteration analyses yield pessimistic failure rates

- ➡ Computing mean time to first failed iteration ignores stability requirements
- ➡ E.g., iteration failure probability of 10^{-10} ↦ **36,000 x 10^{-9} failures / hours**

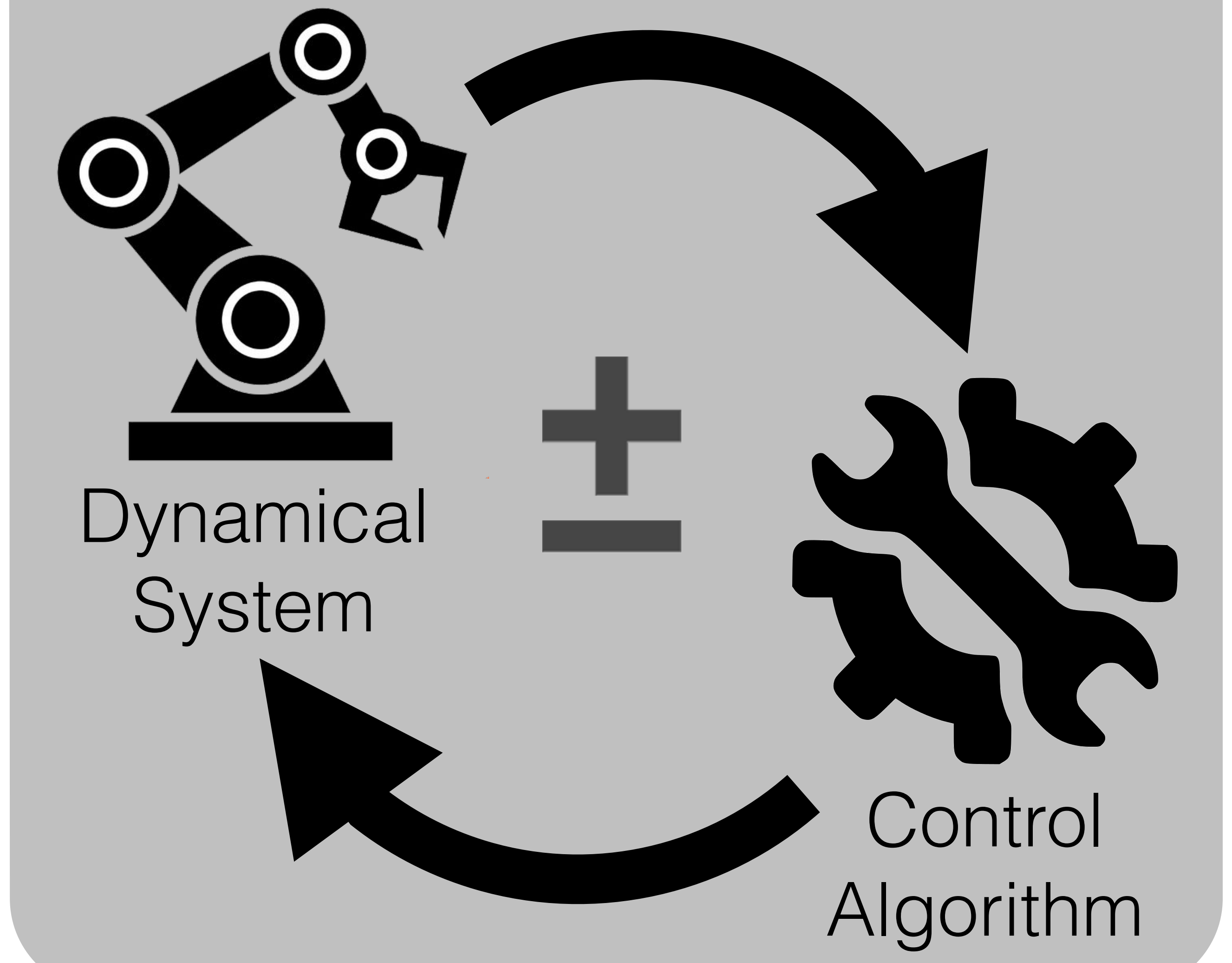
9 orders of magnitude!

Explicitly accounting for the stability requirements

- ➡ Yields more accurate failure rates
- ➡ E.g., iteration failure probability of 10^{-10} and stability requirement ↦ **1.08 x 10^{-15} failures / hours**

How to Analyse the Reliability of **Temporally Robust Systems?**

Control system



Motivating example

- **Frequency:** 100 Hz (10 ms time period)
- **Stability requirement:** 3 out of 4 iterations execute on time
- **Schedulability analyses:** Pr[single iteration delayed] ≤ 10^{-10}

Per-iteration analyses yield pessimistic failure rates

- Computing mean time to first failed iteration ignores stability requirements
- E.g., iteration failure probability of 10^{-10} ↦ **36,000 x 10^{-9} failures / hours**

9 orders of magnitude!

Explicitly accounting for the stability requirements

- Yields more accurate failure rates
- E.g., iteration failure probability of 10^{-10} and stability requirement ↦ **1.08 x 10^{-15} failures / hours**

Not trivial anymore!

This work

How to Analyse the Reliability of **Temporally Robust Systems?**

Objectives

Generic

➡ Complex robustness requirements

How to Analyse the Reliability of **Temporally Robust Systems?**

<u>Objectives</u>

Generic

- ➡ Complex robustness requirements

Accurate (ideally, exact)

- ➡ Minimize pessimism in the final system reliability

How to Analyse the Reliability of **Temporally Robust Systems?**

Objectives

Generic

- ➡ Complex robustness requirements

Accurate (ideally, exact)

- ➡ Minimize pessimism in the final system reliability

Scalable

- ➡ Asymptotic requirements with large parameter values

How to Analyse the Reliability of **Temporally Robust Systems?**

Objectives

Generic
- ➡ Complex robustness requirements

Accurate (ideally, exact)
- ➡ Minimize pessimism in the final system reliability

Scalable
- ➡ Asymptotic requirements with large parameter values

Proposed Techniques

PMC (**P**robabilistic **M**odel **C**hecking)
- ➡ Exact, very generic, but slow

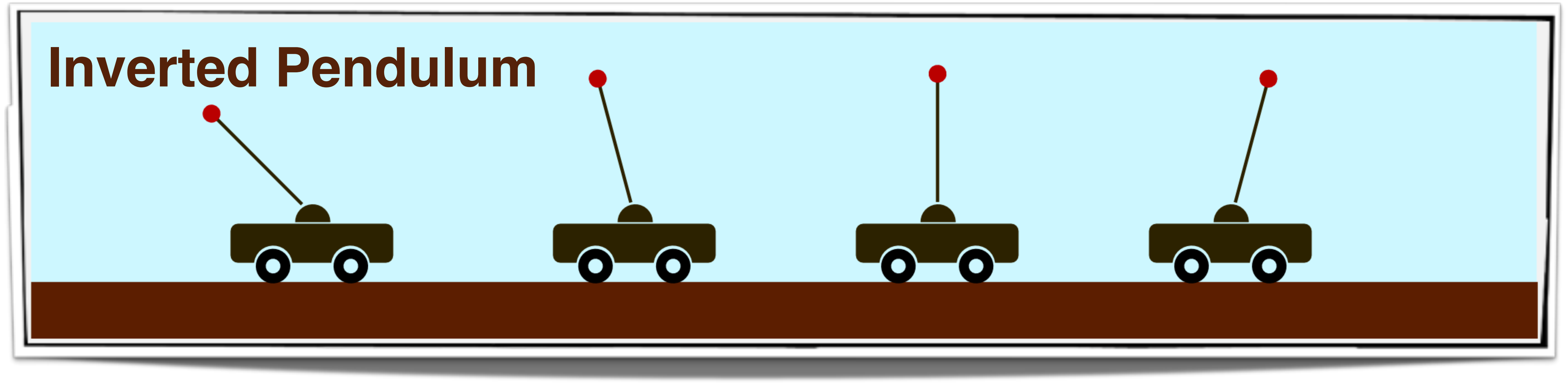
Mart (uses martingale theory)
- ➡ Exact, less generic, but slightly faster

SAp (**S**ound **Ap**proximation)
- ➡ Not exact, least generic, but highly scalable

Background & System Model

Asymptotic Properties

Asymptotic Properties



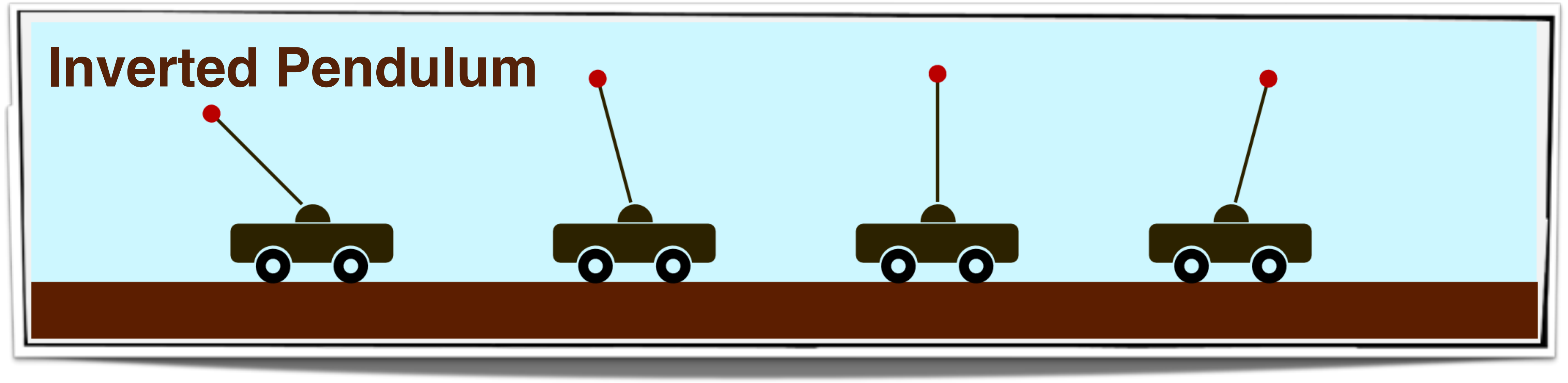
Specification: Mass 0.5 kg, length 0.20 m, period 10 ms

Design: Current iteration is skipped ↦ Use previous iteration parameters

Asymptotically stable with **at least 76.51% successful iterations***

* Majumdar et al. “Performance-aware scheduler synthesis for control systems.” EMSOFT, Taipei (2011)

Asymptotic Properties



Specification: Mass 0.5 kg, length 0.20 m, period 10 ms

Design: Current iteration is skipped ↦ Use previous iteration parameters

Asymptotically stable with **at least 76.51% successful iterations***

Doesn't specify if the system can handle a burst of skipped iterations

➡ What if the first 50 iterations are skipped? No feedback for 0.5 second

* Majumdar et al. "Performance-aware scheduler synthesis for control systems." EMSOFT, Taipei (2011)

Weakly-Hard* Constraints

Concretize asymptotic properties using finite window sizes

* Bernat et al. "Weakly hard real-time systems." *IEEE Transactions on Computers*, 50(4):308–321 (2001).

Weakly-Hard* Constraints

Concretize asymptotic properties using finite window sizes

If each iteration is labeled either as a **S**uccess or a **F**ailure

- ➡ (m, k) constraint: At least m out of every k consecutive iterations must be **S**uccessful

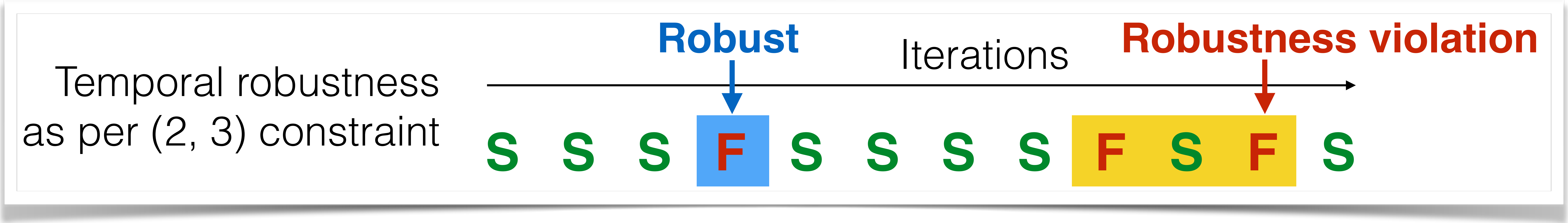
[*] Bernat et al. "Weakly hard real-time systems." *IEEE Transactions on Computers*, 50(4):308–321 (2001).

Weakly-Hard* Constraints

Concretize asymptotic properties using finite window sizes

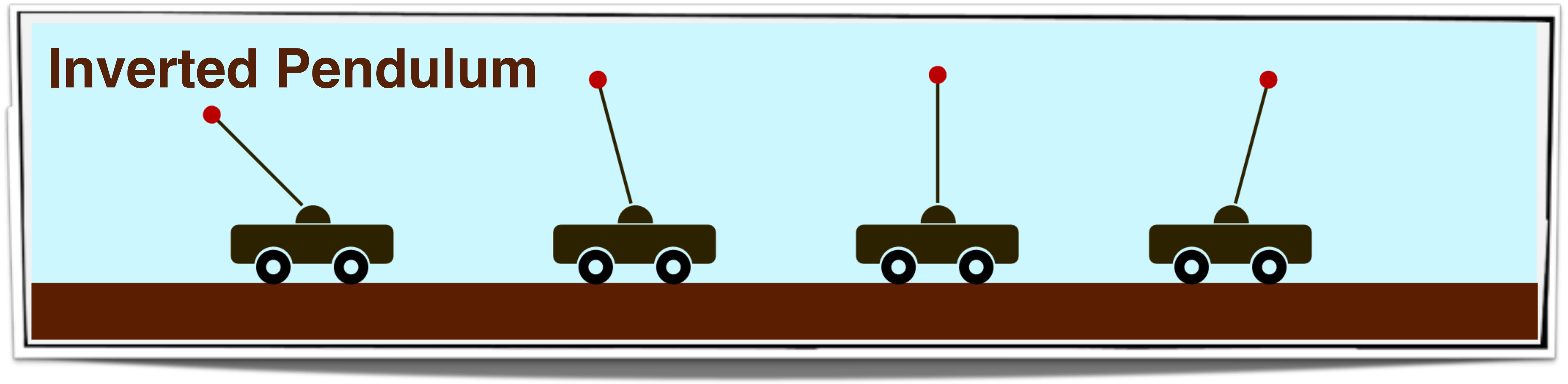
If each iteration is labeled either as a **S**uccess or a **F**ailure

➡ (m, k) constraint: At least m out of every k consecutive iterations must be **S**uccessful



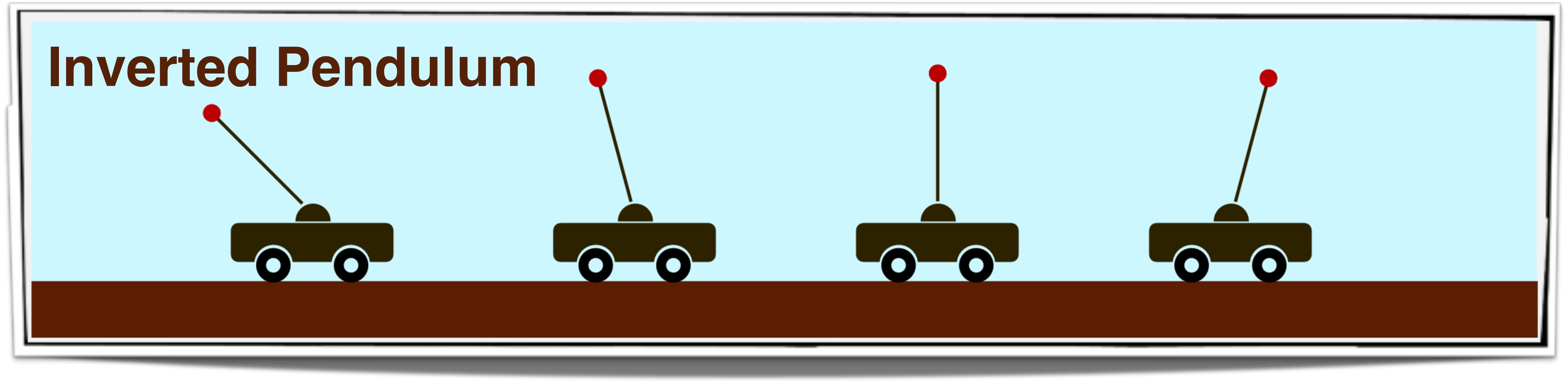
* Bernat et al. "Weakly hard real-time systems." *IEEE Transactions on Computers*, 50(4):308–321 (2001).

Temporal Robustness Criteria



Asymptotically stable with **at least 76.51% successful iterations*** ⟼ (766, 1000)

Temporal Robustness Criteria

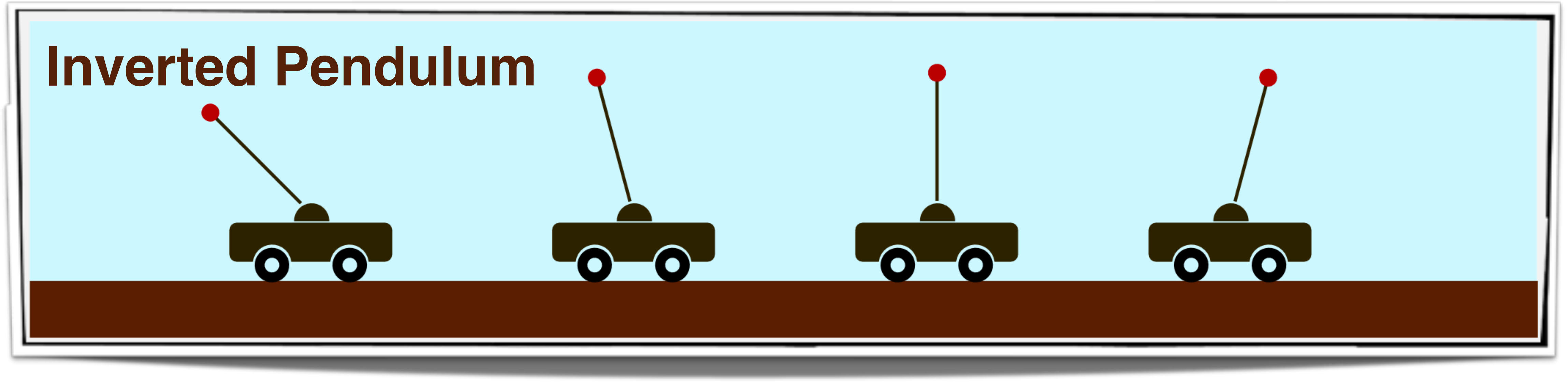


Asymptotically stable with **at least 76.51% successful iterations*** ⟼ (766, 1000)

Short-range “liveness” constraints ⟼ (1, 5)

- ➡ The inverted pendulum can tolerate a small burst of skipped iterations

Temporal Robustness Criteria



Robustness Criteria

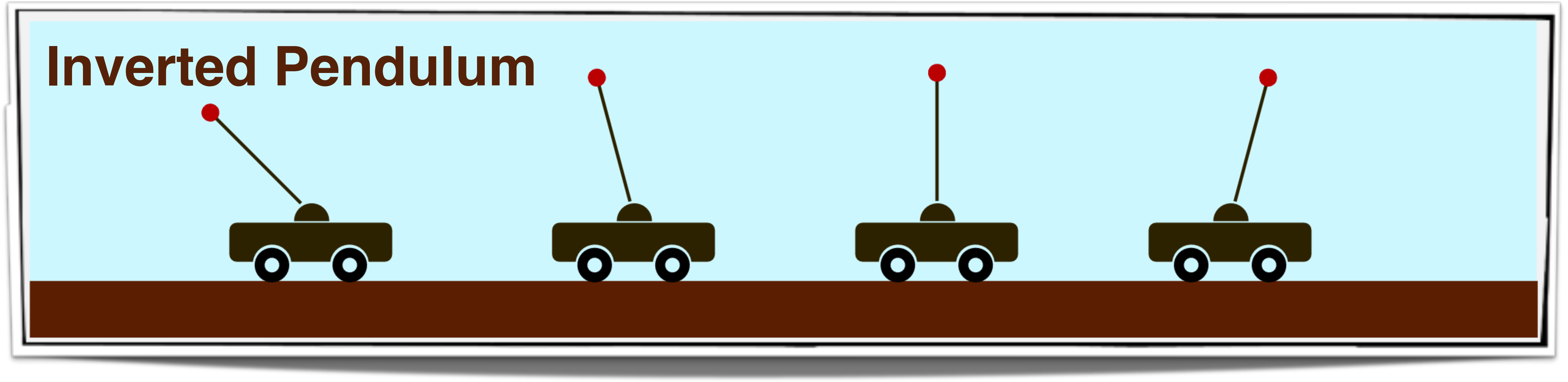
Combination of two weakly-hard constraints

Asymptotically stable with **at least 76.51% successful iterations*** ⟼ (766, 1000)

Short-range "liveness" constraints ⟼ (1, 5)

- ➡ The inverted pendulum can tolerate a small burst of skipped iterations

Temporal Robustness Criteria



Robustness Criteria

Combination of two weakly-hard constraints

Asymptotically stable with **at least 76.51% successful iterations*** ⟼ (766, 1000)

Short-range "liveness" constraints ⟼ (1, 5)

➡ The inverted pendulum can tolerate a small burst of skipped iterations

Combination of different weakly-hard constraints

➡ (m, k) = Each k consecutive iterations, at least m successes needed

➡ $\langle m, k \rangle$ = Each k consecutive iterations, at least m consecutive successes needed

➡ $\overline{\langle m \rangle}$ = m consecutive failures should never happen

Problem Statement

Given periodic system **S**, time period **T**, iteration failure probability $\mathbf{P_F}$, and the **temporal robustness criteria** ...

Problem Statement

Given periodic system **S**, time period **T**, iteration failure probability $\mathbf{P_F}$, and the **temporal robustness criteria** …

Lower-bound the **Mean Time To Failure (MTTF)** of S

$$\text{MTTF} = \text{Expected time to } 1^{st} \text{ temporal robustness violation}$$

$$= \sum_{n=0}^{\infty} \left(nT \times \Pr[1^{st} \text{ violation in the } n^{th} \text{ iteration}] \right)$$

Problem Statement

Given periodic system **S**, time period **T**, iteration failure probability $\mathbf{P_F}$, and the **temporal robustness criteria** ...

Lower-bound the **Mean Time To Failure (MTTF)** of S

MTTF = Expected time to 1st temporal robustness violation

$$= \sum_{n=0}^{\infty} \left(nT \times \Pr[1^{st} \text{ violation in the } n^{th} \text{ iteration}] \right)$$

$$\propto \frac{1}{\text{failure rate (e.g., failures/hour or FIT)}}$$

Problem Statement

Given periodic system **S**, time period **T**, iteration failure probability $\mathbf{P_F}$, and the **temporal robustness criteria** …

Lower-bound the **Mean Time To Failure (MTTF)** of S

$$\text{MTTF} = \text{Expected time to } 1^{st} \text{ temporal robustness violation}$$

$$= \sum_{n=0}^{\infty} \left(nT \times \Pr[1^{st} \text{ violation in the } n^{th} \text{ iteration}] \right)$$

$$\propto \frac{1}{\text{failure rate (e.g., failures/hour or FIT)}}$$

Assumption: $\mathbf{P_F}$ is independently and identically distributed (IID)[1, 2]

[1] Broster et al. "Timing Analysis of Real-Time Communication Under Electromagnetic Interference." Real Time Systems Journal (2005)

[2] Gujarati et al. "Quantifying the Resiliency of Fail-Operational Real-Time Networked Control Systems." ECRTS, Barcelona (2018)

Probabilistic **M**odel **C**hecking **(PMC)**

Exact, very generic, but slow

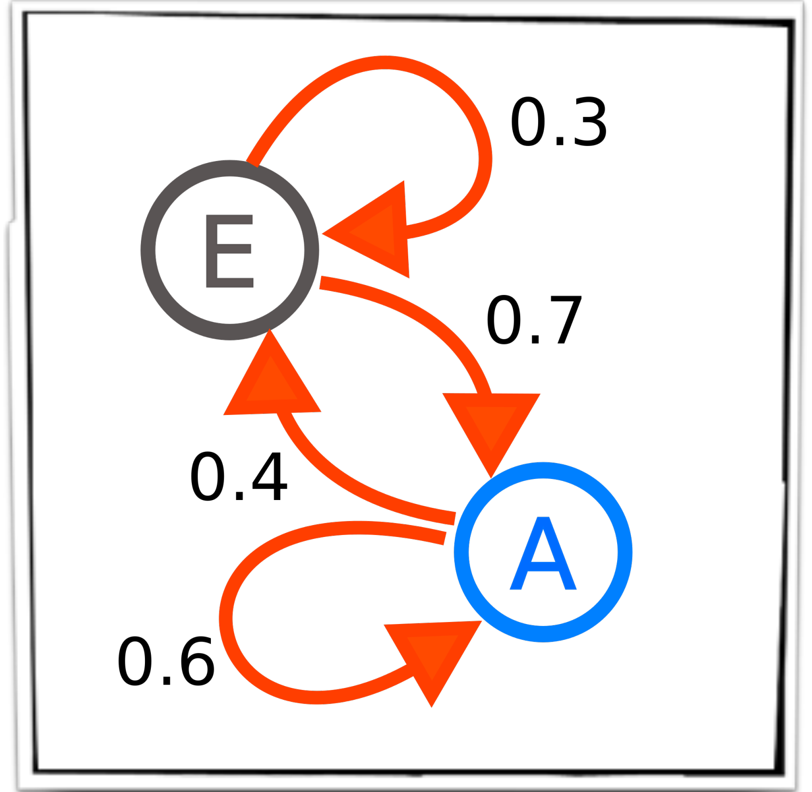
MTTF Estimation using **PMC**

Formal verification technique to model and analyze systems that exhibit **probabilistic** behaviours

MTTF Estimation using **PMC**

Formal verification technique to model and analyze systems that exhibit **probabilistic** behaviours

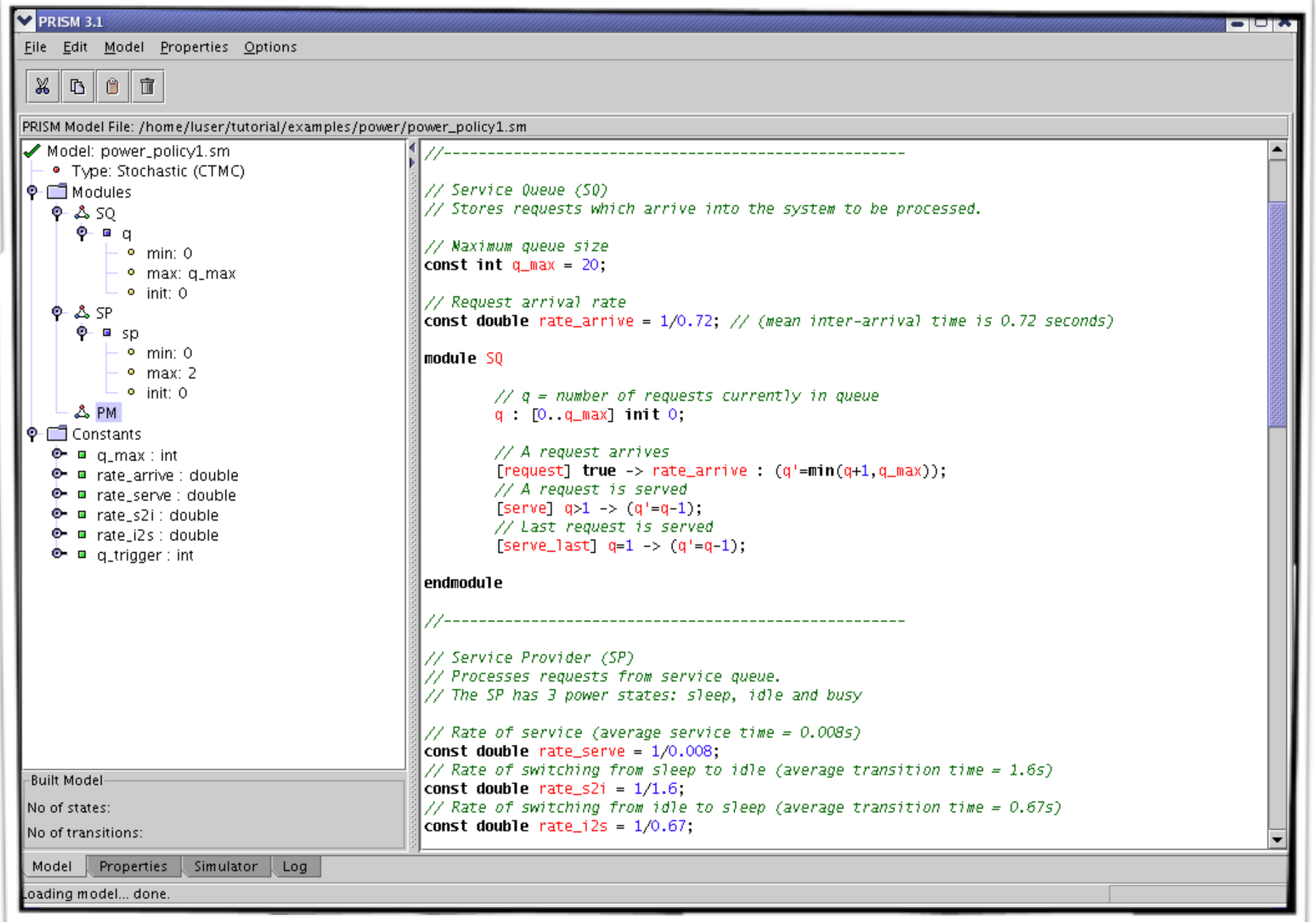
System as a probabilistic model



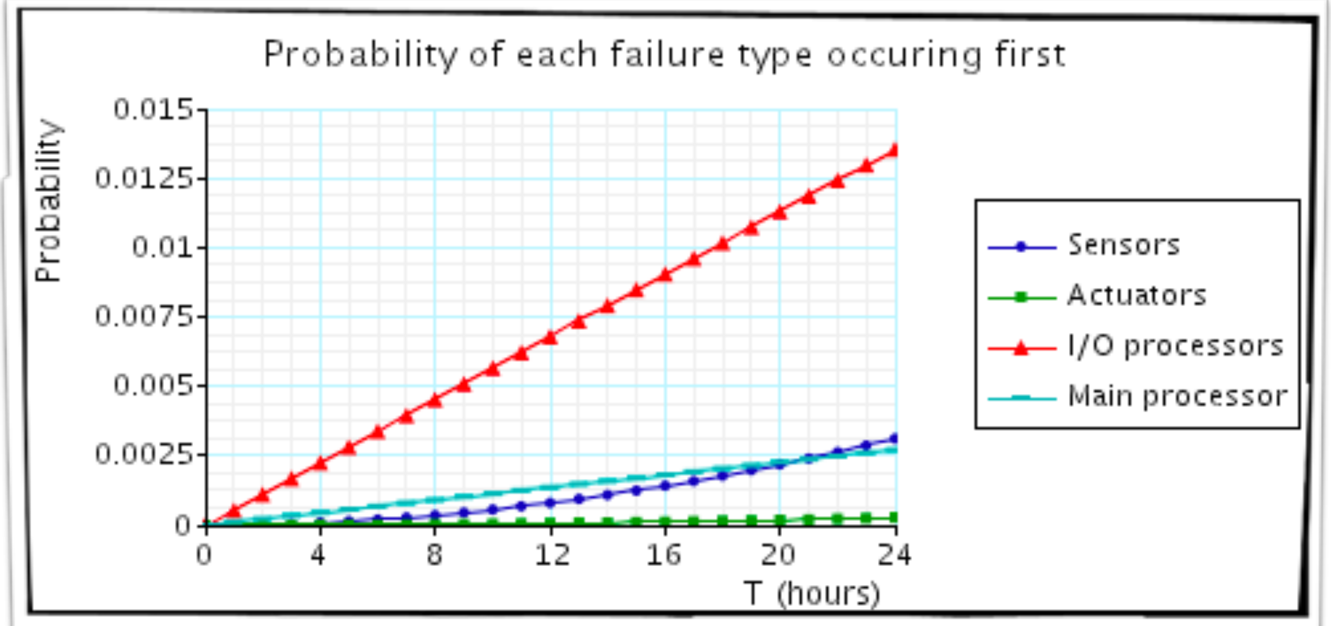
Safety properties as temporal logical specifications

`R=? [ F safe=false ]`

Probabilistic model checker, e.g., PRISM



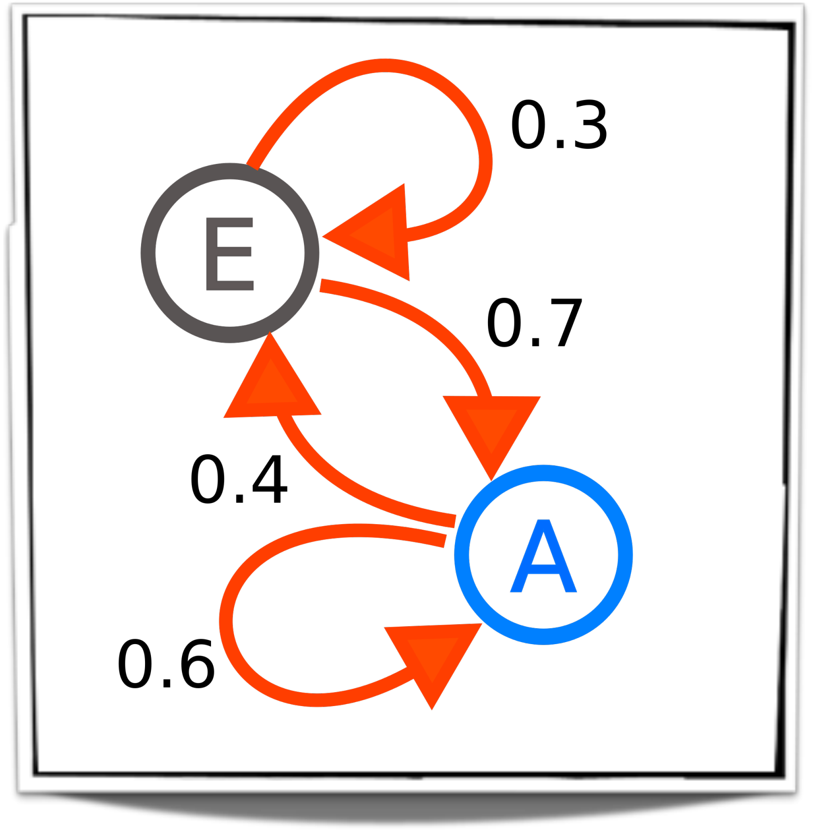
Quantitative results



MTTF Estimation using **PMC**

Periodic system **S** with iteration failure probabilities $\mathbf{P_F}$

System as a probabilistic model



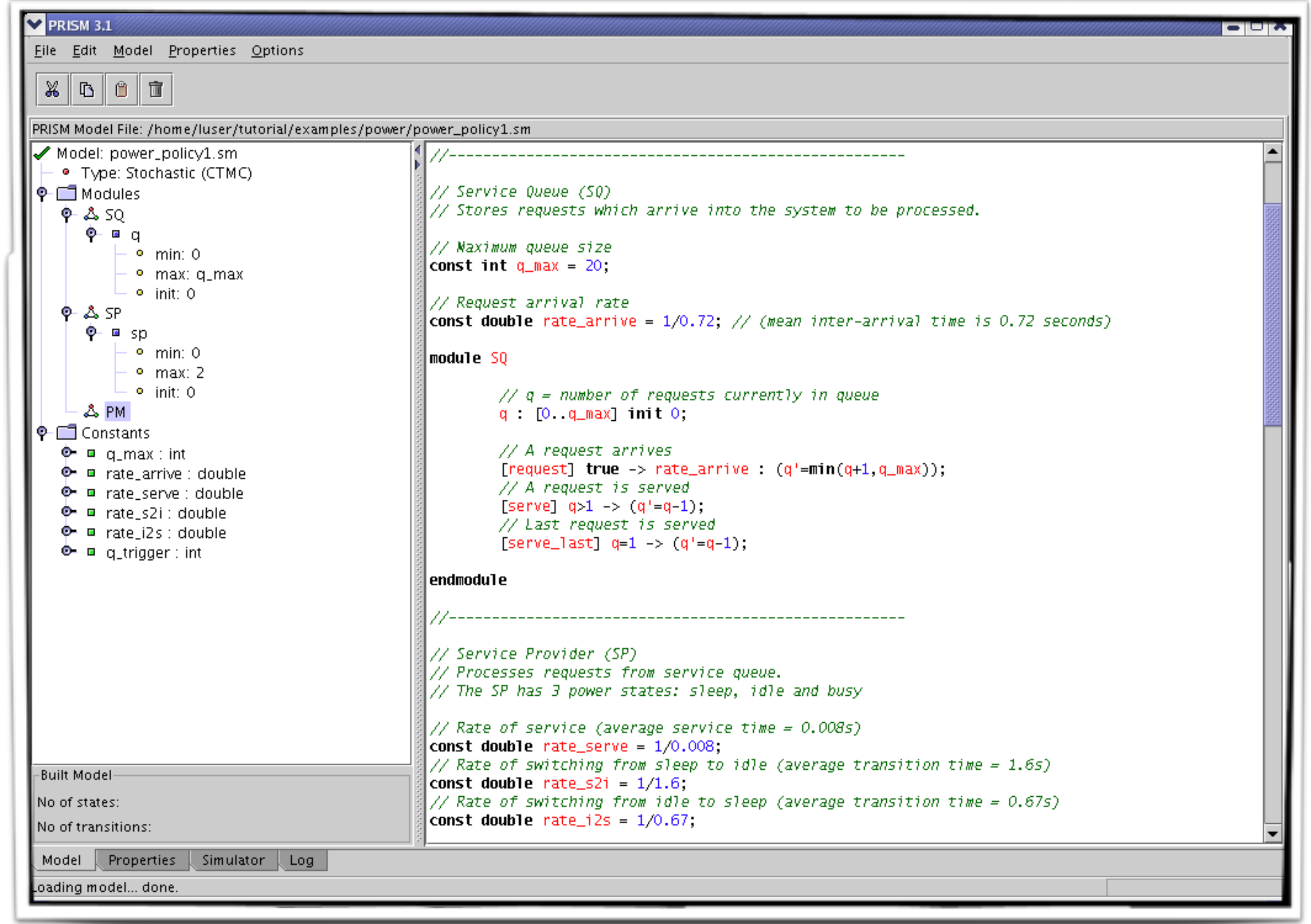
Safety properties as temporal logical specifications

```
R=? [ F safe=false ]
```

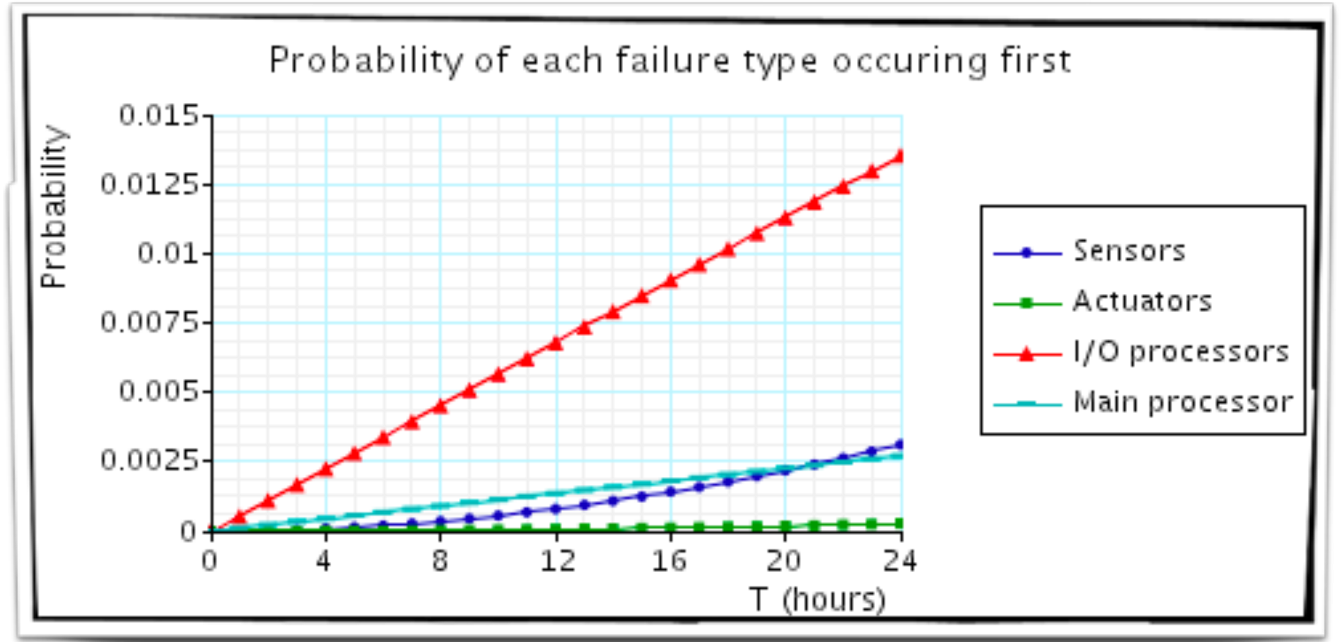
Violation of temporal robustness

Formal verification technique to model and analyze systems that exhibit **probabilistic** behaviours

Probabilistic model checker, e.g., PRISM



Quantitative results



MTTF estimation using PRISM

Modeling Weakly-Hard Constraints

Key idea

Weakly-hard constraints depend on a **finite-sized history**

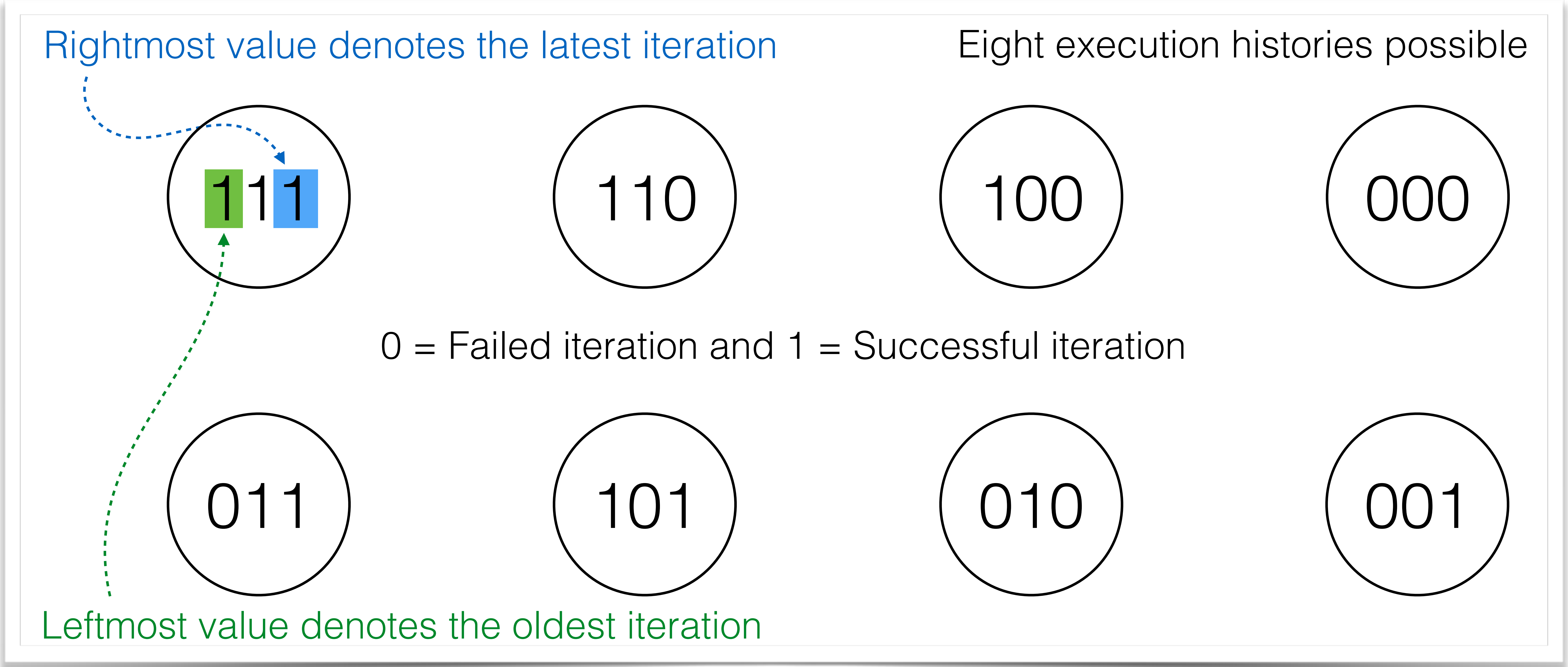
- ➡ E.g., (m, k) constraint depends on the k latest iterations
- ➡ Connect all possible execution histories via transition probabilities $\mathbf{P_F}$ and $\mathbf{1 - P_F}$

Modeling Weakly-Hard Constraints

Key idea

Weakly-hard constraints depend on a **finite-sized history**

- ➡ E.g., (m, k) constraint depends on the k latest iterations
- ➡ Connect all possible execution histories via transition probabilities $\mathbf{P_F}$ and $\mathbf{1 - P_F}$



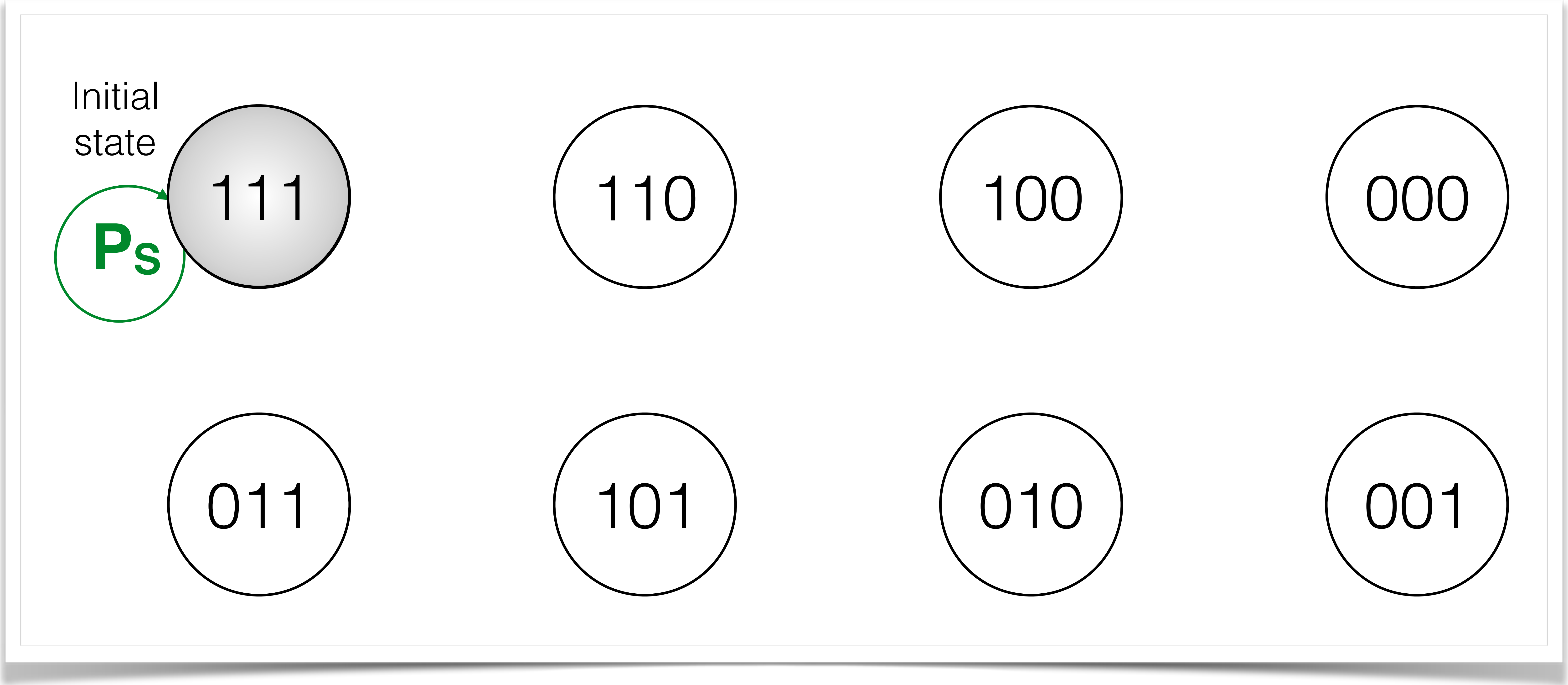
Example:
(2, 3) constraints

Modeling Weakly-Hard Constraints

Key idea

Weakly-hard constraints depend on a **finite-sized history**

- E.g., (m, k) constraint depends on the k latest iterations
- Connect all possible execution histories via transition probabilities $\mathbf{P_F}$ and $\mathbf{1 - P_F}$



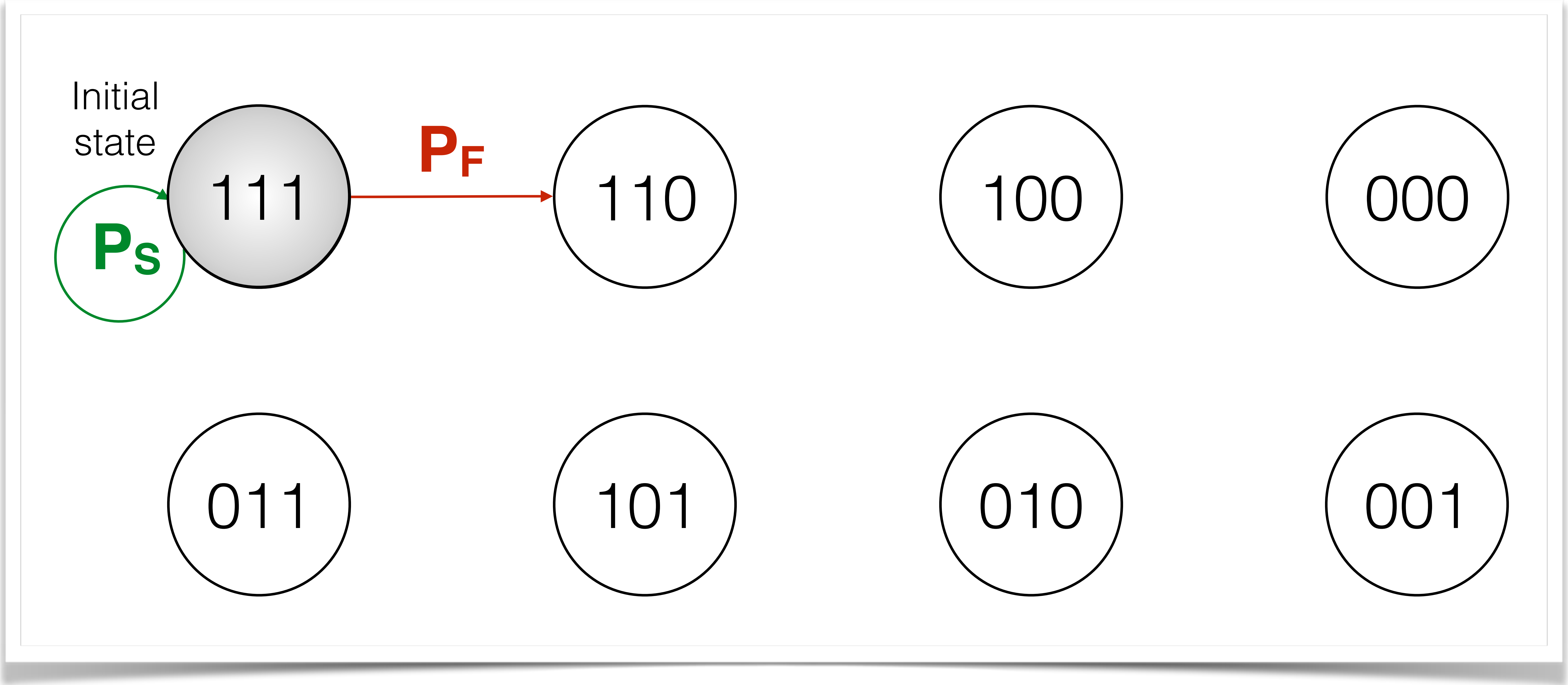
Example:
(2, 3) constraints

Modeling Weakly-Hard Constraints

Key idea

Weakly-hard constraints depend on a **finite-sized history**

- E.g., (m, k) constraint depends on the k latest iterations
- Connect all possible execution histories via transition probabilities P_F and $1 - P_F$



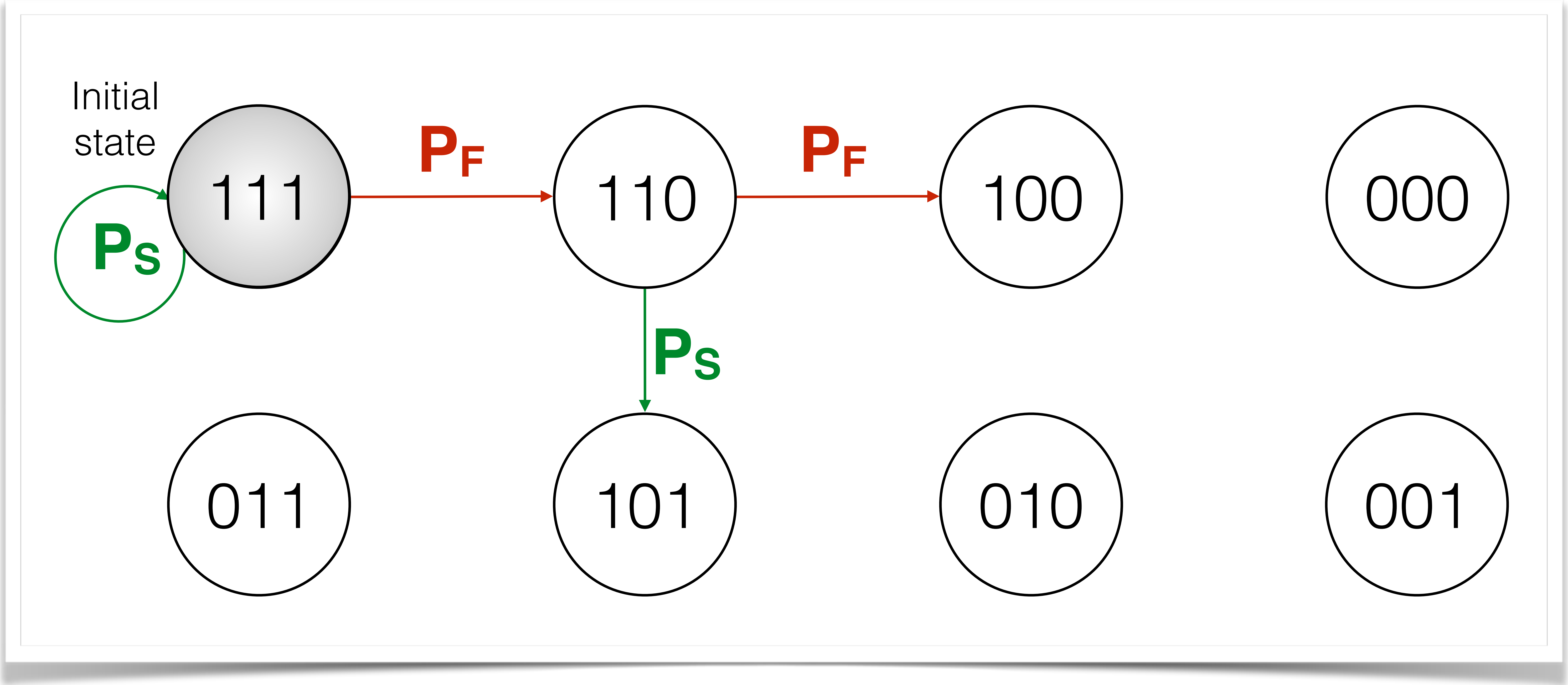
Example:
(2, 3) constraints

Modeling Weakly-Hard Constraints

Key idea

Weakly-hard constraints depend on a **finite-sized history**

- ➡ E.g., (m, k) constraint depends on the k latest iterations
- ➡ Connect all possible execution histories via transition probabilities **P_F** and **$1 - P_F$**



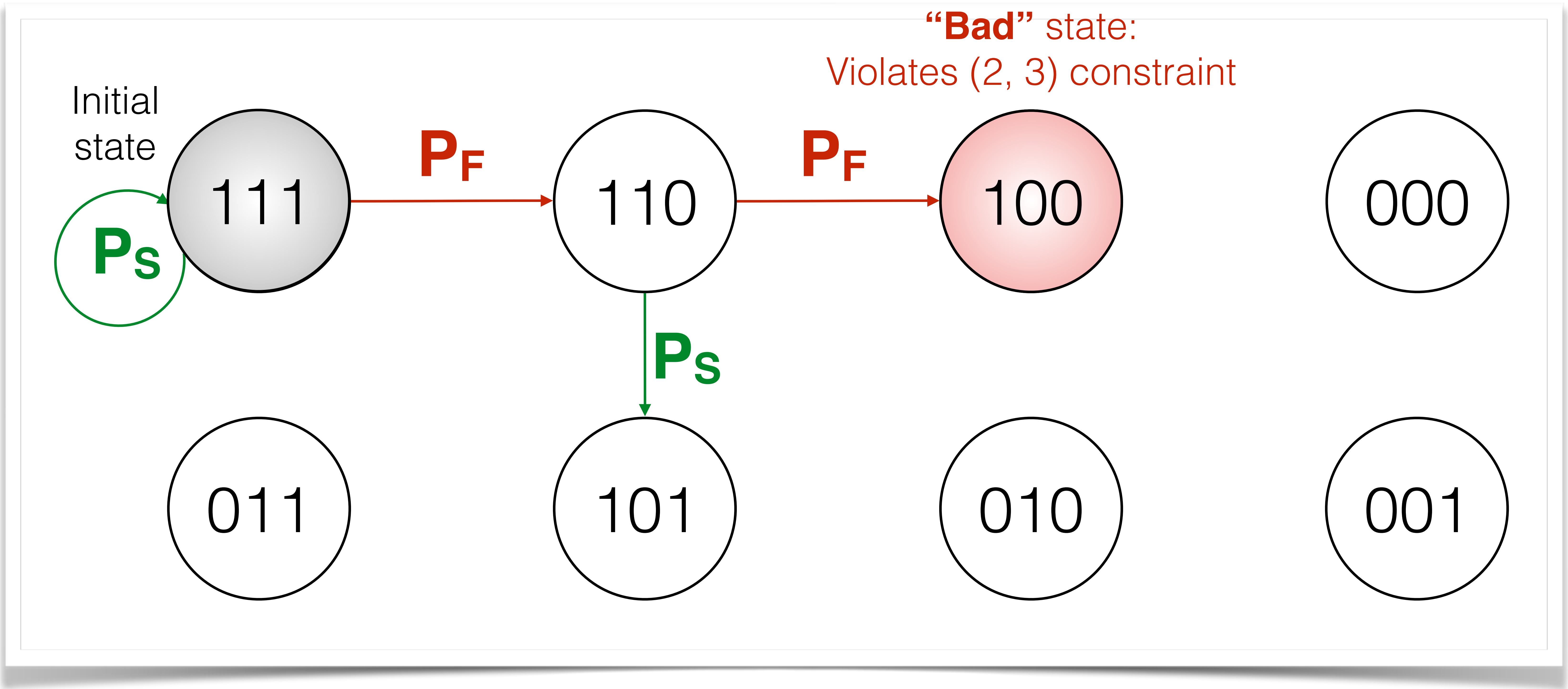
Example:
(2, 3) constraints

Modeling Weakly-Hard Constraints

Key idea

Weakly-hard constraints depend on a **finite-sized history**

- ➡ E.g., (m, k) constraint depends on the k latest iterations
- ➡ Connect all possible execution histories via transition probabilities $\mathbf{P_F}$ and $\mathbf{1 - P_F}$



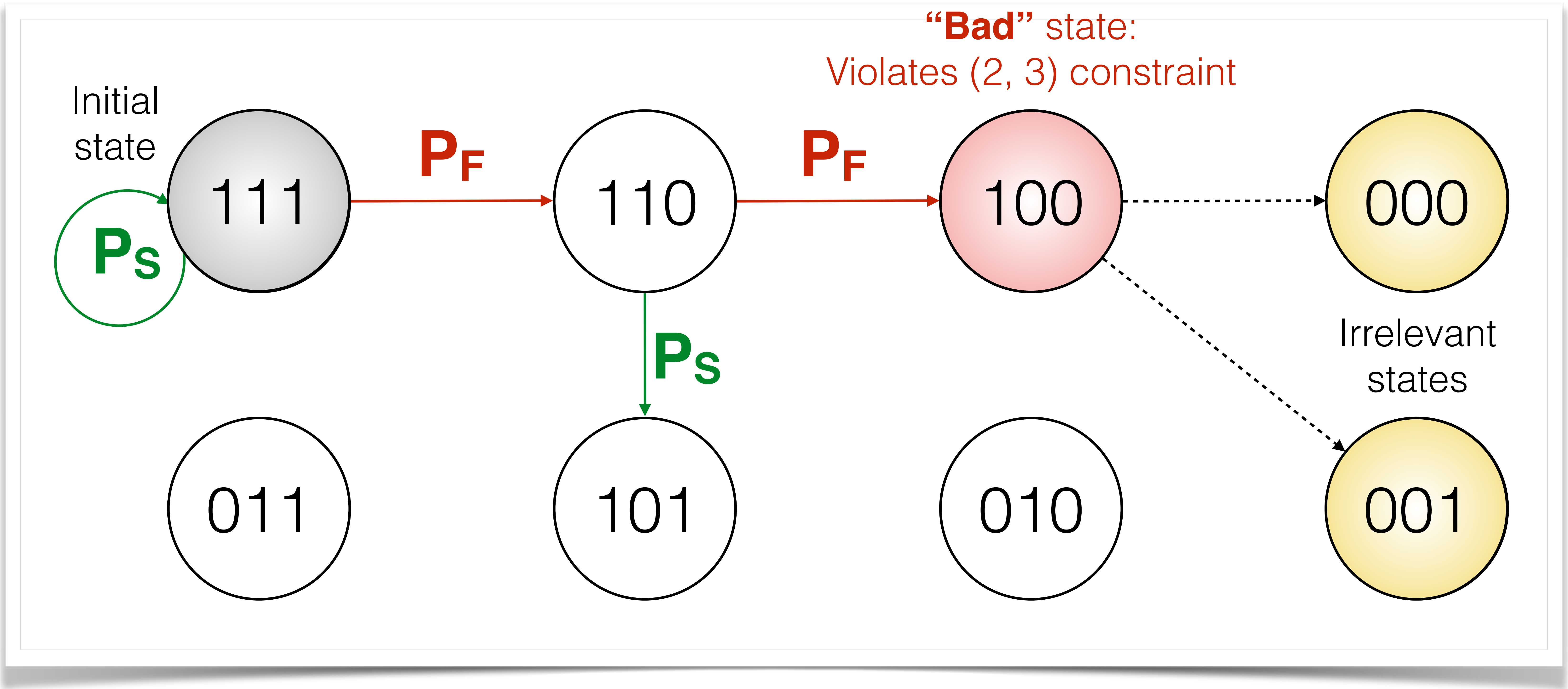
Example:
(2, 3) constraints

Modeling Weakly-Hard Constraints

Key idea

Weakly-hard constraints depend on a **finite-sized history**

- ➡ E.g., (m, k) constraint depends on the k latest iterations
- ➡ Connect all possible execution histories via transition probabilities P_F and $1 - P_F$



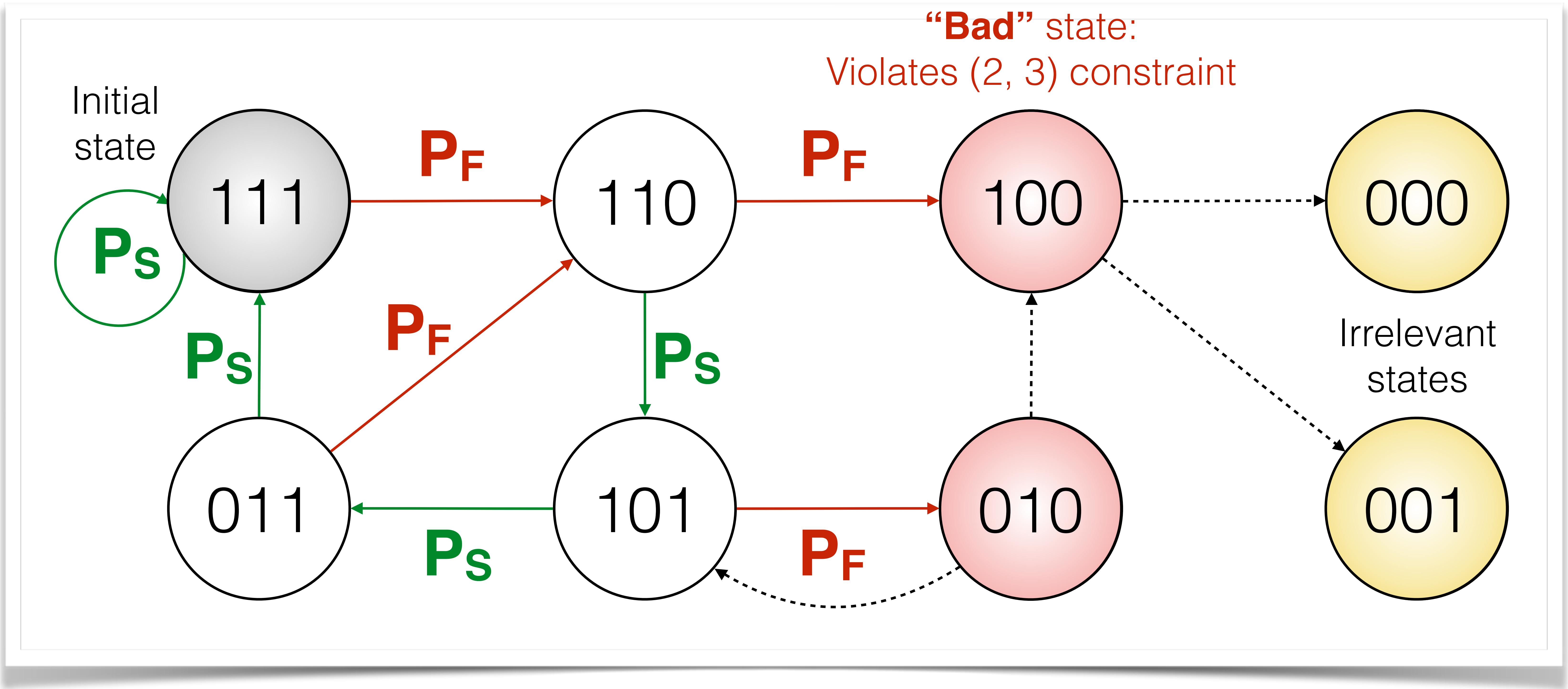
Example:
(2, 3) constraints

Modeling Weakly-Hard Constraints

Key idea

Weakly-hard constraints depend on a **finite-sized history**

- E.g., (m, k) constraint depends on the k latest iterations
- Connect all possible execution histories via transition probabilities **P_F** and **$1 - P_F$**



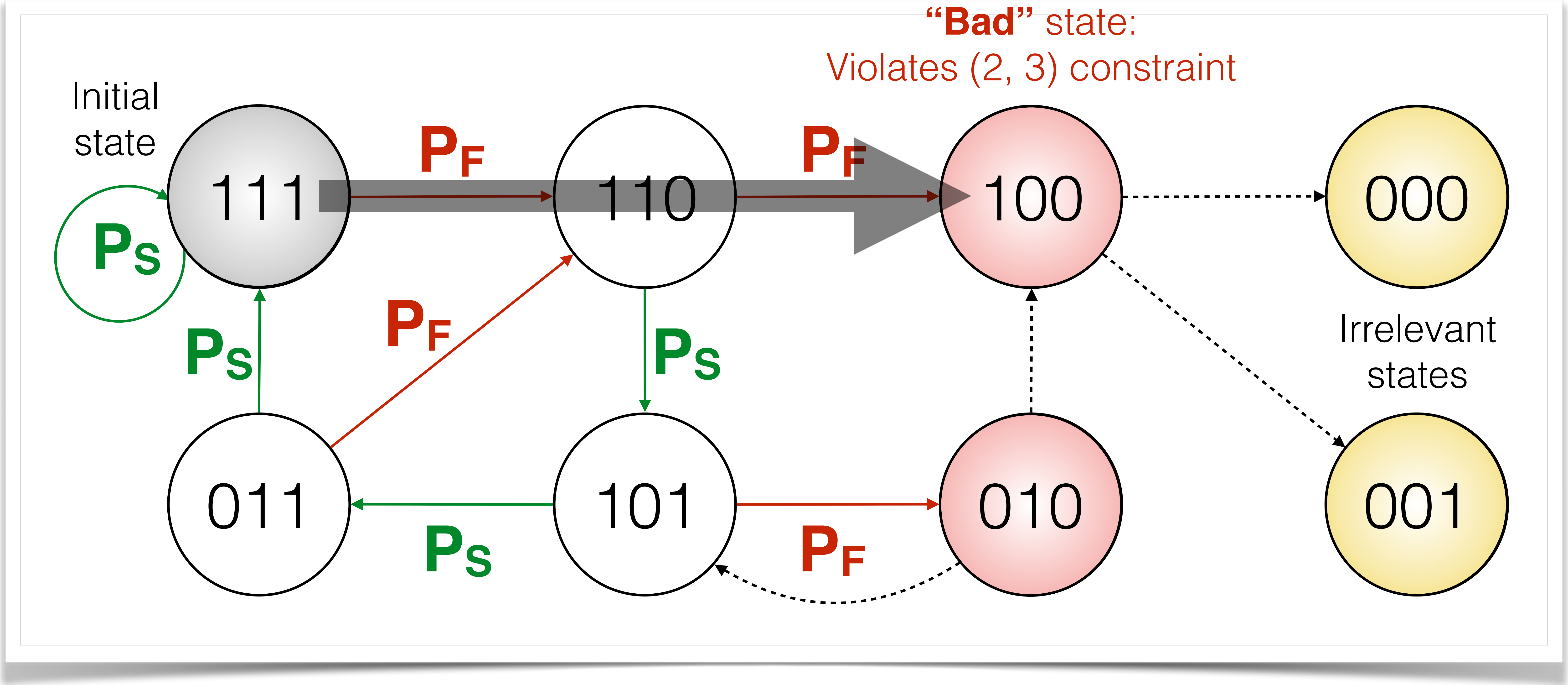
Example:
(2, 3) constraints

Modeling Weakly-Hard Constraints

Key idea

Weakly-hard constraints depend on a **finite-sized history**

- E.g., (m, k) constraint depends on the k latest iterations
- Connect all possible execution histories via transition probabilities **P_F** and **$1 - P_F$**



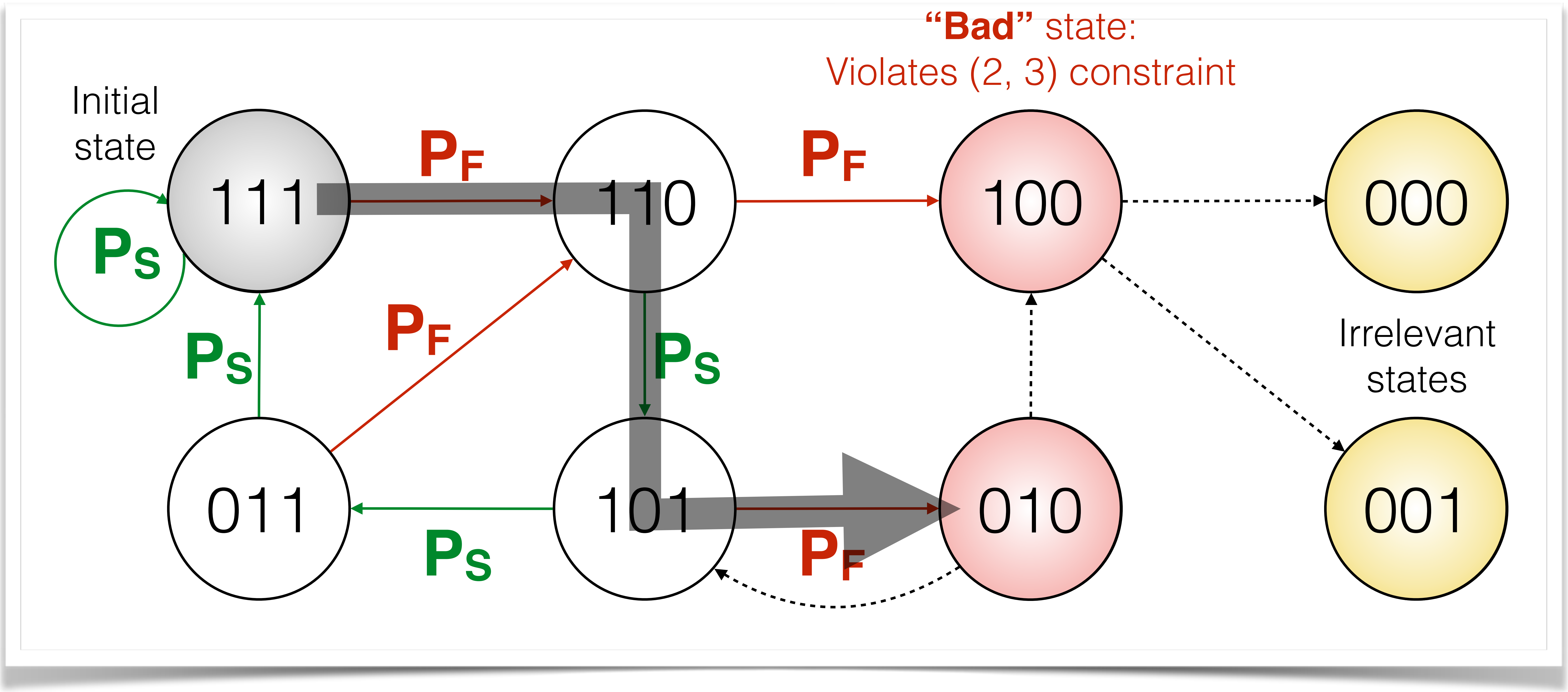
Example:
(2, 3) constraints

Modeling Weakly-Hard Constraints

Key idea

Weakly-hard constraints depend on a **finite-sized history**

- ➡ E.g., (m, k) constraint depends on the k latest iterations
- ➡ Connect all possible execution histories via transition probabilities **P_F** and **1 - P_F**



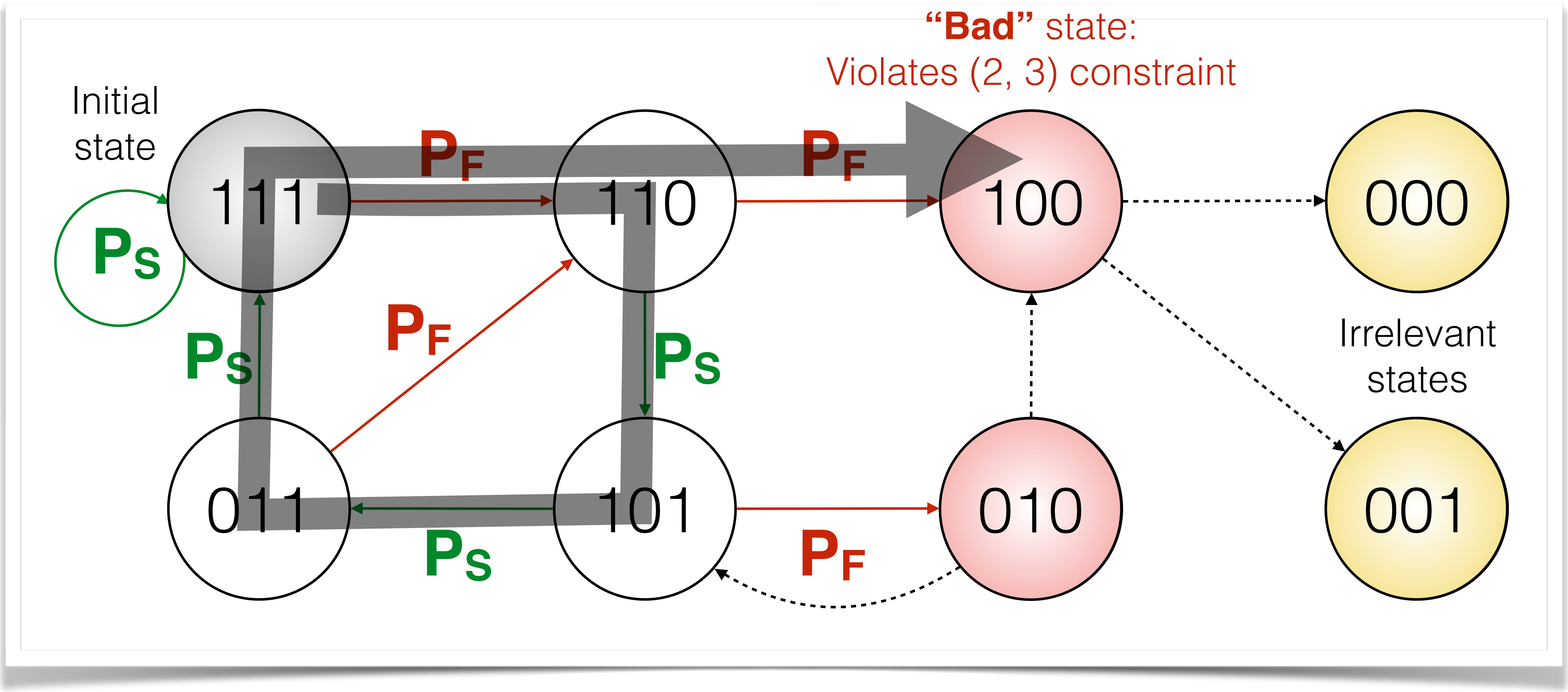
Example:
(2, 3) constraints

Modeling Weakly-Hard Constraints

Key idea

Weakly-hard constraints depend on a **finite-sized history**

- E.g., (m, k) constraint depends on the k latest iterations
- Connect all possible execution histories via transition probabilities **P_F** and **1 - P_F**



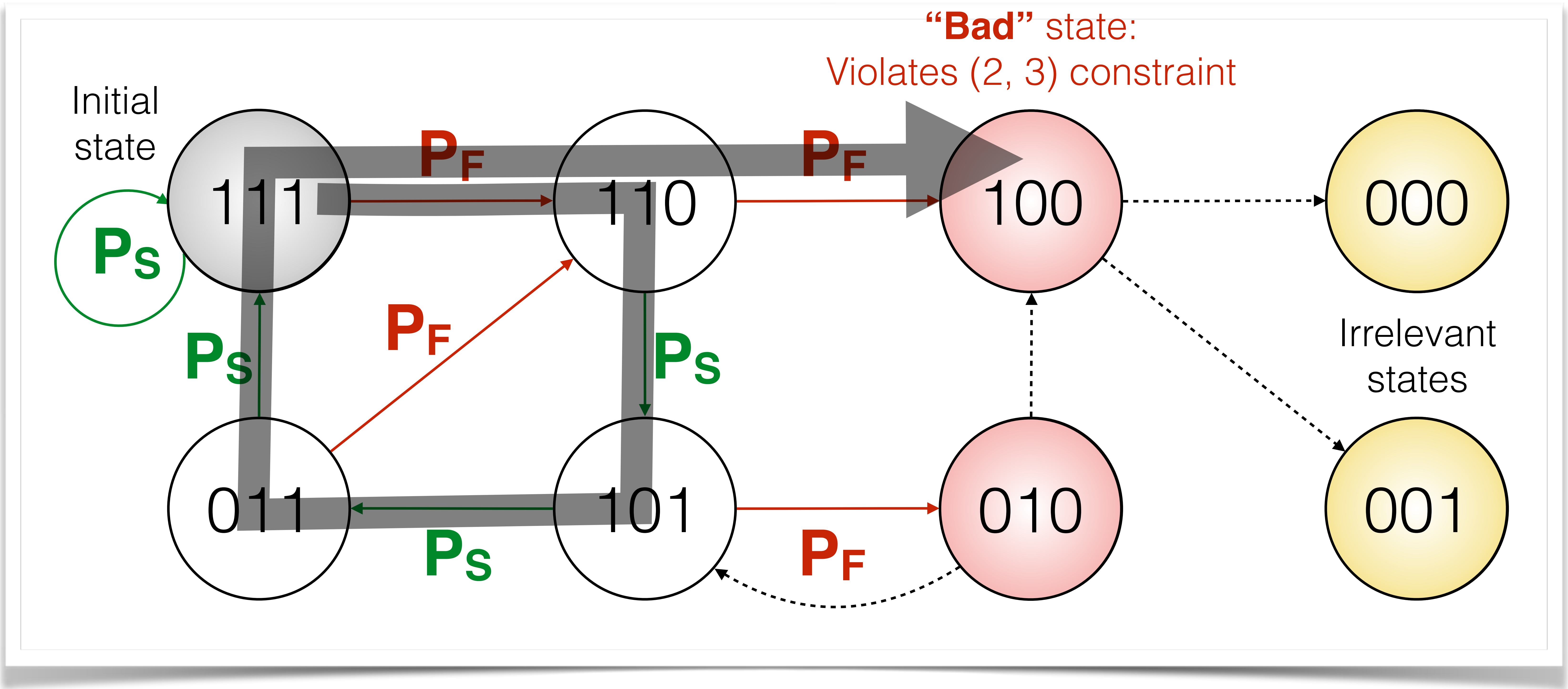
Example:
(2, 3) constraints

Modeling Weakly-Hard Constraints

Key idea

Weakly-hard constraints depend on a **finite-sized history**

- ➡ E.g., (m, k) constraint depends on the k latest iterations
- ➡ Connect all possible execution histories via transition probabilities P_F and $1 - P_F$



Example:
(2, 3) constraints

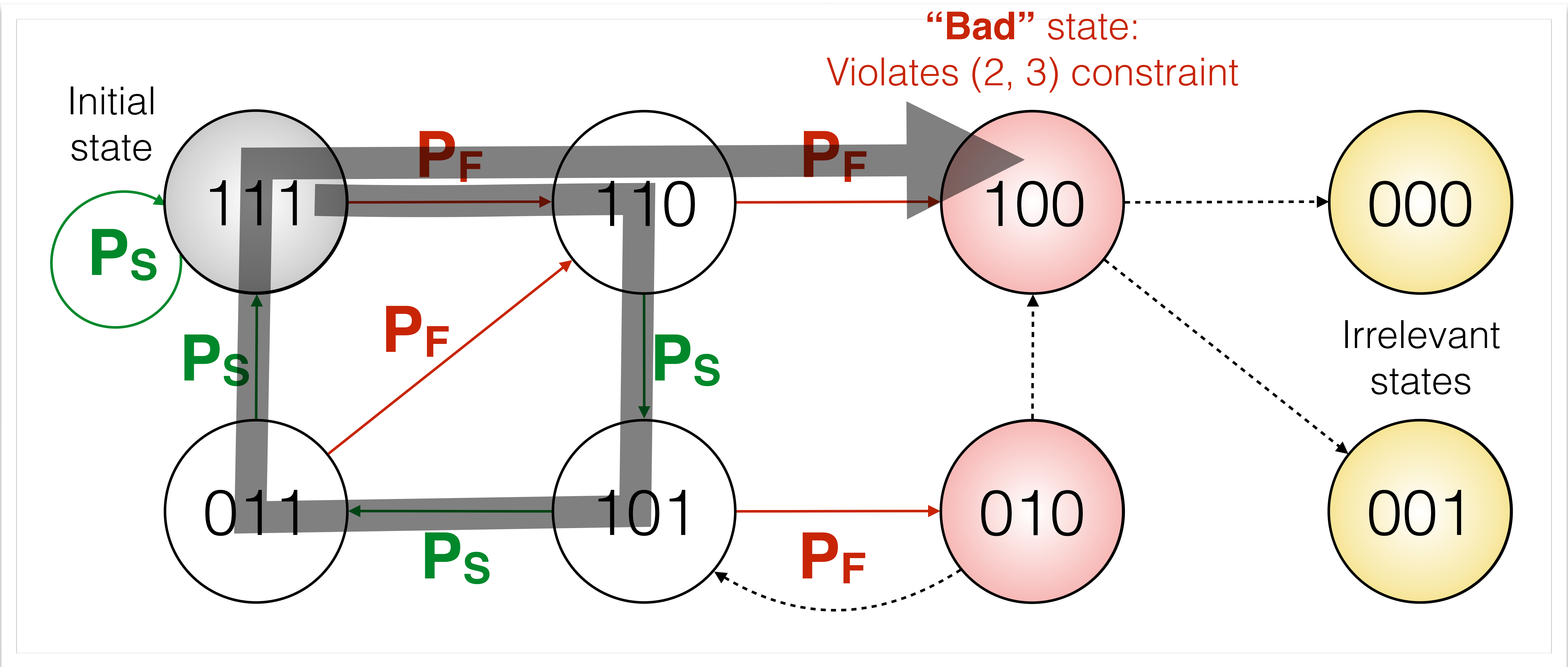
$$MTTF = \left(\text{Expected \# steps to a “bad” state}\right) \times T$$

Modeling Weakly-Hard Constraints

Key idea

Weakly-hard constraints depend on a **finite-sized history**

- ➡ E.g., (m, k) constraint depends on the k latest iterations
- ➡ Connect all possible execution histories via transition probabilities $\mathbf{P_F}$ and $\mathbf{1 - P_F}$



Example:
(2, 3) constraints

$$\text{MTTF} = \underbrace{\left(\begin{array}{c}\text{Expected \# steps}\\ \text{to a “\textbf{bad}” state}\end{array}\right)}_{\text{PRISM}} \times T$$

Does PMC **Scale** with k?

Does PMC **Scale** with k?

P = PMC

24																								?
23																							?	?
22																						?	?	?
21																					?	?	?	?
20																				?	?	?	?	?
19																			?	?	?	?	?	?
18																		?	?	?	?	?	?	?
17																	?	?	?	?	?	?	?	?
16																?	?	?	?	?	?	?	?	?
15															?	?	?	?	?	?	?	?	?	?
14														?	?	?	?	?	?	?	?	?	?	?
13													?	?	?	?	?	?	?	?	?	?	?	?
12												?	?	?	?	?	?	?	?	?	?	?	?	?
11											?	?	?	?	?	?	?	?	?	?	?	?	?	?
10										P	?	?	?	?	?	?	?	?	?	?	?	?	?	?
9									P	P	?	?	?	?	?	?	?	?	?	?	?	?	?	?
8								P	P	P	?	?	?	?	?	?	?	?	?	?	?	?	?	?
7							P	P	P	P	?	?	?	?	?	?	?	?	?	?	?	?	?	?
6						P	P	P	P	P	?	?	?	?	?	?	?	?	?	?	?	?	?	?
5					P	P	P	P	P	P	?	?	?	?	?	?	?	?	?	?	?	?	?	?
4				P	P	P	P	P	P	P	?	?	?	?	?	?	?	?	?	?	?	?	?	?
3			P	P	P	P	P	P	P	P	?	?	?	?	?	?	?	?	?	?	?	?	?	?
↑ 2		P	P	P	P	P	P	P	P	P	?	?	?	?	?	?	?	?	?	?	?	?	?	?
m 1	P	P	P	P	P	P	P	P	P	P	?	?	?	?	?	?	?	?	?	?	?	?	?	?
k →	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Does PMC **Scale** with k?

P = PMC

	24																								?
	23																							?	?
	22																						?	?	?
	21																					?	?	?	?
	20																				?	?	?	?	?
	19																			?	?	?	?	?	?
	18																		?	?	?	?	?	?	?
	17																	?	?	?	?	?	?	?	?
	16																?	?	?	?	?	?	?	?	?
	15															?	?	?	?	?	?	?	?	?	?
	14														?	?	?	?	?	?	?	?	?	?	?
	13													?	?	?	?	?	?	?	?	?	?	?	?
	12												?	?	?	?	?	?	?	?	?	?	?	?	?
	11											?	?	?	?	?	?	?	?	?	?	?	?	?	?
	10										P	?	?	?	?	?	?	?	?	?	?	?	?	?	?
	9									P	P	?	?	?	?	?	?	?	?	?	?	?	?	?	?
	8								P	P	P	?	?	?	?	?	?	?	?	?	?	?	?	?	?
	7							P	P	P	P	?	?	?	?	?	?	?	?	?	?	?	?	?	?
	6						P	P	P	P	P	?	?	?	?	?	?	?	?	?	?	?	?	?	?
	5					P	P	P	P	P	P	?	?	?	?	?	?	?	?	?	?	?	?	?	?
	4				P	P	P	P	P	P	P	?	?	?	?	?	?	?	?	?	?	?	?	?	?
	3			P	P	P	P	P	P	P	P	?	?	?	?	?	?	?	?	?	?	?	?	?	?
↑	2		P	P	P	P	P	P	P	P	P	?	?	?	?	?	?	?	?	?	?	?	?	?	?
m	1	P	P	P	P	P	P	P	P	P	P	?	?	?	?	?	?	?	?	?	?	?	?	?	?
k	→	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

PMC times out after 1 hour for each k > 11

Optimizing for the **Common Case k - m ≪ k**

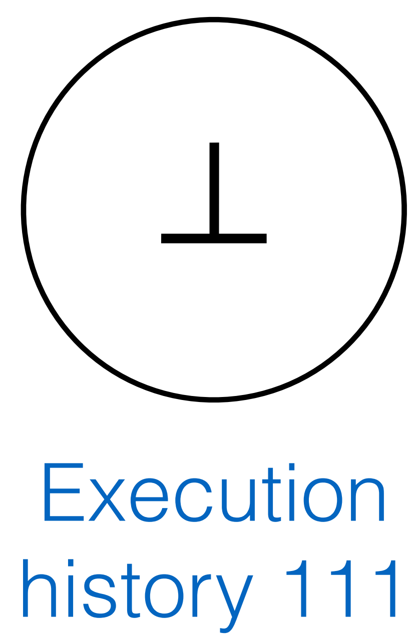
Optimizing for the **Common Case k - m ≪ k**

Store **positions of all failed iterations**, instead of the entire history

Optimizing for the **Common Case k - m ≪ k**

Store **positions of all failed iterations**, instead of the entire history

Example: (2, 3) constraint



Optimizing for the **Common Case k - m ≪ k**

Store **positions of all failed iterations**, instead of the entire history

Example: (2, 3) constraint

$\perp$

Execution history 111

1

Execution history 011

Optimizing for the **Common Case k - m ≪ k**

Store **positions of all failed iterations**, instead of the entire history

Example: (2, 3) constraint

⊥	1	2	3
Execution history 111	Execution history 011	Execution history 101	Execution history 110

Optimizing for the **Common Case k - m ≪ k**

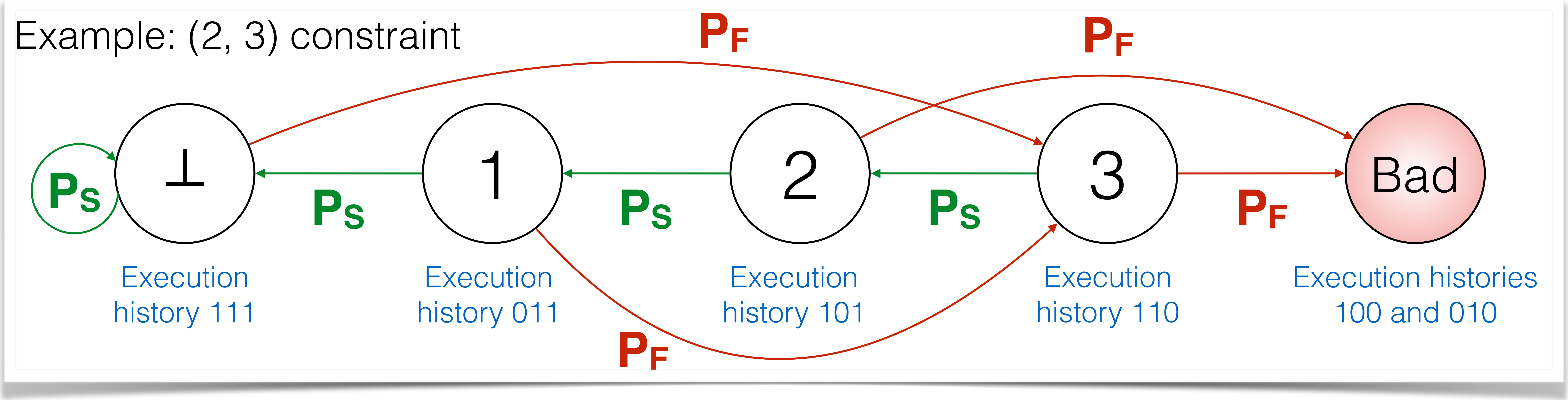
Store **positions of all failed iterations**, instead of the entire history

Example: (2, 3) constraint

⊥	1	2	3	Bad
Execution history 111	Execution history 011	Execution history 101	Execution history 110	Execution histories 100 and 010

Optimizing for the **Common Case k - m ≪ k**

Store **positions of all failed iterations**, instead of the entire history



Does the Optimized PMC **Scale** with k?

P = PMC

24																								P
23																							P	P
22																						P	P	?
21																					P	P	?	?
20																				P	P	?	?	?
19																			P	P	?	?	?	?
18																		P	P	?	?	?	?	?
17																	P	P	?	?	?	?	?	?
16																P	P	?	?	?	?	?	?	?
15															P	P	?	?	?	?	?	?	?	?
14														P	P	P	?	?	?	?	?	?	?	?
13													P	P	P	?	?	?	?	?	?	?	?	?
12												P	P	P	?	?	?	?	?	?	?	?	?	?
11											P	P	P	?	?	?	?	?	?	?	?	?	?	?
10										P	P	P	?	?	?	?	?	?	?	?	?	?	?	?
9									P	P	P	?	?	?	?	?	?	?	?	?	?	?	?	?
8								P	P	P	P	?	?	?	?	?	?	?	?	?	?	?	?	?
7							P	P	P	P	?	?	?	?	?	?	?	?	?	?	?	?	?	?
6						P	P	P	P	P	?	?	?	?	?	?	?	?	?	?	?	?	?	?
5					P	P	P	P	P	P	?	?	?	?	?	?	?	?	?	?	?	?	?	?
4				P	P	P	P	P	P	P	P	?	?	?	?	?	?	?	?	?	?	?	?	?
3			P	P	P	P	P	P	P	P	P	P	P	P	P	?	?	?	?	?	?	?	?	?
↑ 2		P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
m 1	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
k →	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Does the Optimized PMC **Scale** with k?

P = PMC

24																								P
23																							P	P
22																						P	P	?
21																					P	P	?	?
20																				P	P	?	?	?
19																			P	P	?	?	?	?
18																		P	P	?	?	?	?	?
17																	P	P	?	?	?	?	?	?
16																P	P	?	?	?	?	?	?	?
15															P	P	?	?	?	?	?	?	?	?
14														P	P	P	?	?	?	?	?	?	?	?
13													P	P	P	?	?	?	?	?	?	?	?	?
12												P	P	P	?	?	?	?	?	?	?	?	?	?
11											P	P	P	?	?	?	?	?	?	?	?	?	?	?
10										P	P	P	?	?	?	?	?	?	?	?	?	?	?	?
9									P	P	P	?	?	?	?	?	?	?	?	?	?	?	?	?
8								P	P	P	P	?	?	?	?	?	?	?	?	?	?	?	?	?
7							P	P	P	P	?	?	?	?	?	?	?	?	?	?	?	?	?	?
6						P	P	P	P	P	?	?	?	?	?	?	?	?	?	?	?	?	?	?
5					P	P	P	P	P	P	?	?	?	?	?	?	?	?	?	?	?	?	?	?
4				P	P	P	P	P	P	P	P	?	?	?	?	?	?	?	?	?	?	?	?	?
3			P	P	P	P	P	P	P	P	P	P	P	P	P	?	?	?	?	?	?	?	?	?
↑ 2		P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
m 1	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
k →	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

PMC scales for large k
if m ≪ k or k - m ≪ k

Does the Optimized PMC **Scale** with k?

P = PMC

24																								P
23																							P	P
22																						P	P	?
21																					P	P	?	?
20																				P	P	?	?	?
19																			P	P	?	?	?	?
18																		P	P	?	?	?	?	?
17																	P	P	?	?	?	?	?	?
16																P	P	?	?	?	?	?	?	?
15															P	P	?	?	?	?	?	?	?	?
14														P	P	P	?	?	?	?	?	?	?	?
13													P	P	P	?	?	?	?	?	?	?	?	?
12												P	P	P	?	?	?	?	?	?	?	?	?	?
11											P	P	P	?	?	?	?	?	?	?	?	?	?	?
10										P	P	P	?	?	?	?	?	?	?	?	?	?	?	?
9									P	P	P	?	?	?	?	?	?	?	?	?	?	?	?	?
8								P	P	P	P	?	?	?	?	?	?	?	?	?	?	?	?	?
7							P	P	P	P	?	?	?	?	?	?	?	?	?	?	?	?	?	?
6						P	P	P	P	P	?	?	?	?	?	?	?	?	?	?	?	?	?	?
5					P	P	P	P	P	P	?	?	?	?	?	?	?	?	?	?	?	?	?	?
4				P	P	P	P	P	P	P	P	?	?	?	?	?	?	?	?	?	?	?	?	?
3			P	P	P	P	P	P	P	P	P	P	P	P	P	?	?	?	?	?	?	?	?	?
↑ 2		P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
m 1	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
k →	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

PMC scales for large k
if m « k or k - m « k

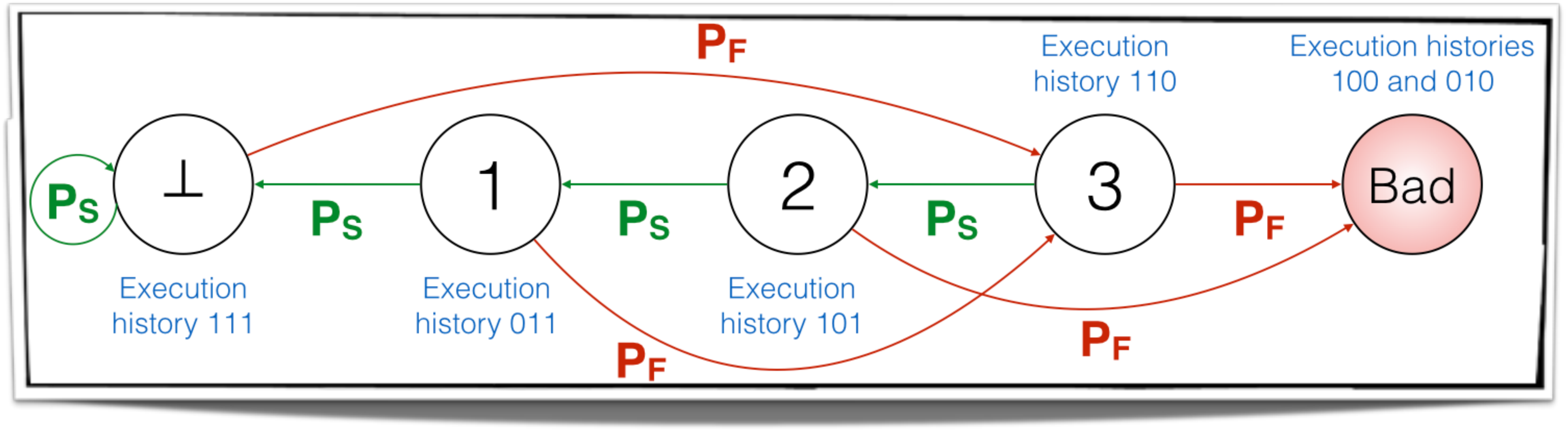
Scalability still a problem
for the general case

The Martingale Approach **(Mart)**

Exact, less generic, but slightly faster

Exact Model Checking Slows Down PRISM

Markov model



Model building

System of linear equations

$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ -1 & 110 & 10 & 110 \\ -1 & 10 & 1090/9 & 10 \\ -1 & 0 & 100/9 & 1000/9 \end{bmatrix} \begin{bmatrix} e_0 \\ e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Model solving

MTTF (expected reward)

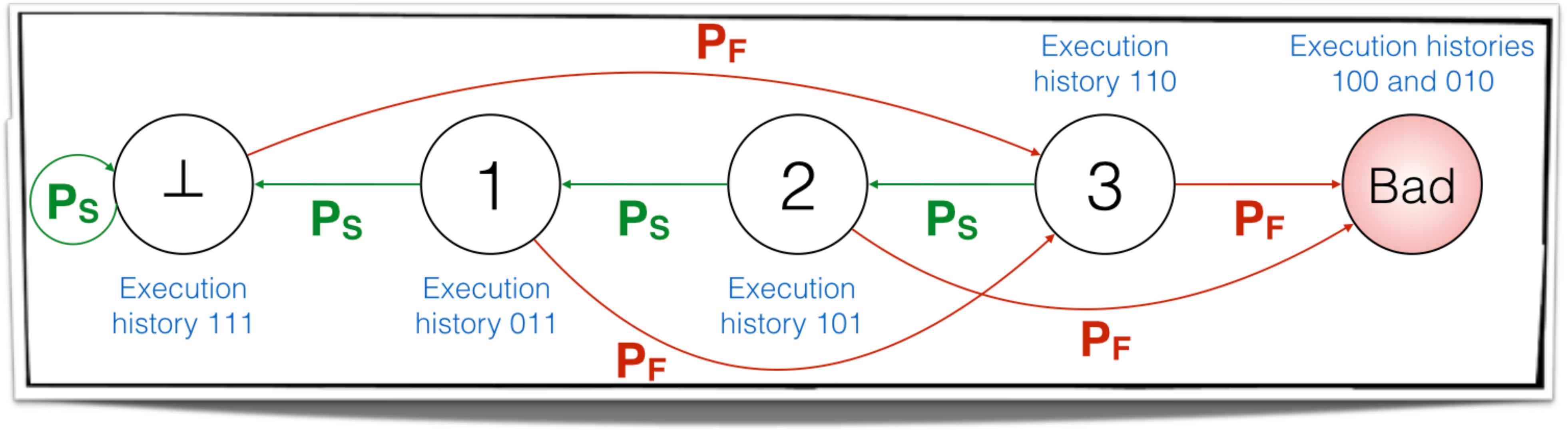
Probabilistic model checking
(PRISM under the hood)

For error-free computation

PRISM must be configured with **exact model checking** (i.e., no floating points)

Exact Model Checking Slows Down PRISM

Markov model



System of linear equations

Model building

$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ -1 & 110 & 10 & 110 \\ -1 & 10 & 1090/9 & 10 \\ -1 & 0 & 100/9 & 1000/9 \end{bmatrix} \begin{bmatrix} e_0 \\ e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Model solving

MTTF (expected reward)

Probabilistic model checking
(PRISM under the hood)

For error-free computation

PRISM must be configured with **exact model checking** (i.e., no floating points)

Using **martingale theory***

- ➡ Linear equations obtained directly
- ➡ Bypass PRISM, use highly-scalable BLAS/LAPACK libraries, with very high precision

* Li. “A Martingale Approach to the Study of Occurrence of Sequence Patterns in Repeated Experiments.” The Annals of Probability 8.6 (1980):1171–1176.

Mart Scales Better than PMC

P = PMC

M = Mart

Mart used only if PMC timed out

24																								P
23																							P	P
22																						P	P	?
21																					P	P	?	?
20																				P	P	?	?	?
19																			P	P	?	?	?	?
18																		P	P	M	?	?	?	?
17																	P	P	M	?	?	?	?	?
16																P	P	M	?	?	?	?	?	?
15															P	P	M	?	?	?	?	?	?	?
14														P	P	P	?	?	?	?	?	?	?	?
13													P	P	P	M	?	?	?	?	?	?	?	?
12												P	P	P	M	M	?	?	?	?	?	?	?	?
11											P	P	P	M	M	?	?	?	?	?	?	?	?	?
10										P	P	P	M	M	M	?	?	?	?	?	?	?	?	?
9									P	P	P	M	M	M	M	?	?	?	?	?	?	?	?	?
8								P	P	P	P	M	M	M	M	?	?	?	?	?	?	?	?	?
7							P	P	P	P	M	M	M	M	M	?	?	?	?	?	?	?	?	?
6						P	P	P	P	P	M	M	M	M	M	M	?	?	?	?	?	?	?	?
5					P	P	P	P	P	P	M	M	M	M	M	M	M	M	M	?	?	?	?	?
4				P	P	P	P	P	P	P	P	M	M	M	M	M	M	M	M	M	M	M	M	M
3			P	P	P	P	P	P	P	P	P	P	P	P	P	M	M	M	M	M	M	M	M	M
↑ 2		P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
m 1	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
k →	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Mart Scales Better than PMC

m ↑ \ k →	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
24																								P
23																							P	P
22																						P	P	?
21																					P	P	?	?
20																				P	P	?	?	?
19																			P	P	?	?	?	?
18																		P	P	M	?	?	?	?
17																	P	P	M	?	?	?	?	?
16																P	P	M	?	?	?	?	?	?
15															P	P	M	?	?	?	?	?	?	?
14														P	P	P	?	?	?	?	?	?	?	?
13													P	P	P	M	?	?	?	?	?	?	?	?
12												P	P	P	M	M	?	?	?	?	?	?	?	?
11											P	P	P	M	M	?	?	?	?	?	?	?	?	?
10										P	P	P	M	M	M	?	?	?	?	?	?	?	?	?
9									P	P	P	M	M	M	M	?	?	?	?	?	?	?	?	?
8								P	P	P	P	M	M	M	M	?	?	?	?	?	?	?	?	?
7							P	P	P	P	M	M	M	M	M	?	?	?	?	?	?	?	?	?
6						P	P	P	P	P	M	M	M	M	M	M	?	?	?	?	?	?	?	?
5					P	P	P	P	P	P	M	M	M	M	M	M	M	M	M	?	?	?	?	?
4				P	P	P	P	P	P	P	P	M	M	M	M	M	M	M	M	M	M	M	M	M
3			P	P	P	P	P	P	P	P	P	P	P	P	P	M	M	M	M	M	M	M	M	M
2		P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
1	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P

P = PMC

M = Mart

Mart used only if PMC timed out

Mart helps scale up exact MTTF estimation to **k = 16**

Mart Scales Better than PMC

m \ k	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
24																								P
23																							P	P
22																						P	P	?
21																					P	P	?	?
20																				P	P	?	?	?
19																			P	P	?	?	?	?
18																		P	P	M	?	?	?	?
17																	P	P	M	?	?	?	?	?
16																P	P	M	?	?	?	?	?	?
15															P	P	M	?	?	?	?	?	?	?
14														P	P	P	?	?	?	?	?	?	?	?
13													P	P	P	M	?	?	?	?	?	?	?	?
12												P	P	P	M	M	?	?	?	?	?	?	?	?
11											P	P	P	M	M	?	?	?	?	?	?	?	?	?
10										P	P	P	M	M	M	?	?	?	?	?	?	?	?	?
9									P	P	P	M	M	M	M	?	?	?	?	?	?	?	?	?
8								P	P	P	P	M	M	M	M	?	?	?	?	?	?	?	?	?
7							P	P	P	P	M	M	M	M	M	?	?	?	?	?	?	?	?	?
6						P	P	P	P	P	M	M	M	M	M	M	?	?	?	?	?	?	?	?
5					P	P	P	P	P	P	M	M	M	M	M	M	M	M	M	?	?	?	?	?
4				P	P	P	P	P	P	P	P	M	M	M	M	M	M	M	M	M	M	M	M	M
3			P	P	P	P	P	P	P	P	P	P	P	P	P	M	M	M	M	M	M	M	M	M
2		P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
1	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P

P = PMC

M = Mart

Mart used only if PMC timed out

Mart helps scale up exact MTTF estimation to **k = 16**

Also, Mart implicitly benefits from small values of m

Mart Scales Better than PMC

P = PMC

M = Mart

Mart used only if PMC timed out

	24																								P
	23																							P	P
	22																						P	P	?
	21																					P	P	?	?
	20																				P	P	?	?	?
	19																			P	P	?	?	?	?
	18																		P	P	M	?	?	?	?
	17																	P	P	M	?	?	?	?	?
	16																P	P	M	?	?	?	?	?	?
	15															P	P	M	?	?	?	?	?	?	?
	14														P	P	P	?	?	?	?	?	?	?	?
	13													P	P	P	M	?	?	?	?	?	?	?	?
	12												P	P	P	M	M	?	?	?	?	?	?	?	?
	11											P	P	P	M	M	?	?	?	?	?	?	?	?	?
	10										P	P	P	M	M	M	?	?	?	?	?	?	?	?	?
	9									P	P	P	M	M	M	M	?	?	?	?	?	?	?	?	?
	8								P	P	P	P	M	M	M	M	?	?	?	?	?	?	?	?	?
	7							P	P	P	P	M	M	M	M	M	?	?	?	?	?	?	?	?	?
	6						P	P	P	P	P	M	M	M	M	M	M	?	?	?	?	?	?	?	?
	5					P	P	P	P	P	P	M	M	M	M	M	M	M	M	M	?	?	?	?	?
	4				P	P	P	P	P	P	P	P	M	M	M	M	M	M	M	M	M	M	M	M	M
	3			P	P	P	P	P	P	P	P	P	P	P	P	P	M	M	M	M	M	M	M	M	M
↑	2		P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
m	1	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
k	→	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Mart helps scale up exact MTTF estimation to **k = 16**

Also, Mart implicitly benefits from small values of m

Scalability still a problem for the general case

Sound **Ap**proximation **(SAp)**

Not exact, least generic, but highly scalable

Sound Approximation (SAp) for Single (m, k) Constraint

Sound Approximation (SAp) for Single (m, k) Constraint

MTTF = Expected time to 1st temporal robustness violation

$$= \sum_{n=0}^{\infty} \Big(nT \times \Pr[1^{st} \text{ violation in the } n^{th} \text{ iteration}] \Big)$$

Sound Approximation (SAp) for Single (m, k) Constraint

MTTF = Expected time to 1st temporal robustness violation

$$= \sum_{n=0}^{\infty}\left(nT \times \underbrace{\Pr[1^{st} \text{ violation in the } n^{th} \text{ iteration}]}_{f(n)}\right)$$

Sound Approximation (SAp) for Single (m, k) Constraint

MTTF = Expected time to 1st temporal robustness violation

$$= \sum_{n=0}^{\infty} \left(nT \times \underbrace{\Pr[1^{st} \text{ violation in the } n^{th} \text{ iteration}]}_{f(n)} \right)$$

MTTF_LB

① Obtain **$f_{LB}(n) \leq f(n)$** that can be quickly computed for large n

② Compute **$f_{LB}(n_0), f_{LB}(n_1), \ldots, f_{LB}(n_D)$**

③ **Numerically integrate** over subintervals $(n_0, n_1], \ldots, (n_{D-1}, n_D]$

Sound Approximation (SAp) for Single (m, k) Constraint

MTTF = Expected time to 1st temporal robustness violation

$$= \sum_{n=0}^{\infty} \left(nT \times \Pr[1^{st} \text{ violation in the } n^{th} \text{ iteration}] \right)$$

f(n)

Approximation accuracy

- ➡ Accuracy of $f_{LB}(n)$ (reliability modeling literature*)
- ➡ The choice of $n_0, n_1, n_2, \ldots, n_D$ (heuristics based on $f_{LB}(n)$'s shape)

MTTF_{LB}

① Obtain $\mathbf{f_{LB}(n) \leq f(n)}$ that can be quickly computed for large n

② Compute $\mathbf{f_{LB}(n_0), f_{LB}(n_1), \ldots, f_{LB}(n_D)}$

③ **Numerically integrate** over subintervals $(n_0, n_1], \ldots, (n_{D-1}, n_D]$

* Sfakianakis et al.. "Reliability of a consecutive k-out-of-r-from-n: F system." IEEE Transactions on Reliability 41.3 (1992): 442-447.

SAp is Scalable to Very Large Window Sizes

SAp used only if both
PMC and Mart timed out

P = PMC
M = Mart
S = SAp

m ↑ / k →	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
24																								P
23																							P	P
22																						P	P	S
21																					P	P	S	S
20																				P	P	S	S	S
19																			P	P	S	S	S	S
18																		P	P	M	S	S	S	S
17																	P	P	M	S	S	S	S	S
16																P	P	M	S	S	S	S	S	S
15															P	P	M	S	S	S	S	S	S	S
14														P	P	P	S	S	S	S	S	S	S	S
13													P	P	P	M	S	S	S	S	S	S	S	S
12												P	P	P	M	M	S	S	S	S	S	S	S	S
11											P	P	P	M	M	S	S	S	S	S	S	S	S	S
10										P	P	P	M	M	M	S	S	S	S	S	S	S	S	S
9									P	P	P	M	M	M	M	S	S	S	S	S	S	S	S	S
8								P	P	P	P	M	M	M	M	S	S	S	S	S	S	S	S	S
7							P	P	P	P	M	M	M	M	M	S	S	S	S	S	S	S	S	S
6						P	P	P	P	P	M	M	M	M	M	M	S	S	S	S	S	S	S	S
5					P	P	P	P	P	P	M	M	M	M	M	M	M	M	M	S	S	S	S	S
4				P	P	P	P	P	P	P	P	M	M	M	M	M	M	M	M	M	M	M	M	M
3			P	P	P	P	P	P	P	P	P	P	P	P	P	M	M	M	M	M	M	M	M	M
2		P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
1	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P

SAp is Scalable to Very Large Window Sizes

SAp used only if both
PMC and Mart timed out

P = PMC
M = Mart
S = SAp

24																								P
23																							P	P
22																						P	P	S
21																					P	P	S	S
20																				P	P	S	S	S
19																			P	P	S	S	S	S
18																		P	P	M	S	S	S	S
17																	P	P	M	S	S	S	S	S
16																P	P	M	S	S	S	S	S	S
15															P	P	M	S	S	S	S	S	S	S
14														P	P	P	S	S	S	S	S	S	S	S
13													P	P	P	M	S	S	S	S	S	S	S	S
12												P	P	P	M	M	S	S	S	S	S	S	S	S
11											P	P	P	M	M	S	S	S	S	S	S	S	S	S
10										P	P	P	M	M	M	S	S	S	S	S	S	S	S	S
9									P	P	P	M	M	M	M	S	S	S	S	S	S	S	S	S
8								P	P	P	P	M	M	M	M	S	S	S	S	S	S	S	S	S
7							P	P	P	P	M	M	M	M	M	S	S	S	S	S	S	S	S	S
6						P	P	P	P	P	M	M	M	M	M	M	S	S	S	S	S	S	S	S
5					P	P	P	P	P	P	M	M	M	M	M	M	M	M	M	S	S	S	S	S
4				P	P	P	P	P	P	P	P	M	M	M	M	M	M	M	M	M	M	M	M	M
3			P	P	P	P	P	P	P	P	P	P	P	P	P	M	M	M	M	M	M	M	M	M
↑ 2		P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
m 1	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
k →	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

SAp comfortably scales for windows of size k = 1000

How **Accurate** is SAp?

All errors are positive (SAp is proven to under-approximate the exact MTTF)

- Error > 50 %
- 25 % < Error ≤ 50 %
- Error ≤ 25 %

11											81.45
10										79.25	69.20
9									76.61	65.85	58.25
8								73.38	62.01	54.73	47.79
7							69.36	57.75	50.31	44.30	39.06
6						64.29	52.99	44.91	40.11	35.35	31.43
5					57.84	47.34	39.12	34.28	31.22	28.13	25.06
4				49.62	39.96	33.44	28.17	24.82	22.98	21.64	20.28
3			39.29	30.21	25.73	22.68	20.09	17.80	15.93	14.55	13.63
↑ 2		25.91	19.44	15.77	13.60	12.30	11.50	10.98	10.62	10.35	10.13
m 1	05.76	05.76	05.76	05.76	05.76	05.76	05.76	05.76	05.76	05.76	05.76
k →	2	3	4	5	6	7	8	9	10	11	12

How **Accurate** is SAp?

All errors are positive (SAp is proven to under-approximate the exact MTTF)

- Error > 50 %
- 25 % < Error ≤ 50 %
- Error ≤ 25 %

m ↑ \ k →	2	3	4	5	6	7	8	9	10	11	12
11											81.45
10										79.25	69.20
9									76.61	65.85	58.25
8								73.38	62.01	54.73	47.79
7							69.36	57.75	50.31	44.30	39.06
6						64.29	52.99	44.91	40.11	35.35	31.43
5					57.84	47.34	39.12	34.28	31.22	28.13	25.06
4				49.62	39.96	33.44	28.17	24.82	22.98	21.64	20.28
3			39.29	30.21	25.73	22.68	20.09	17.80	15.93	14.55	13.63
2		25.91	19.44	15.77	13.60	12.30	11.50	10.98	10.62	10.35	10.13
1	05.76	05.76	05.76	05.76	05.76	05.76	05.76	05.76	05.76	05.76	05.76

Relative errors significant even for small k

- ➡ Exact analysis needed when feasible

How **Accurate** is SAp?

All errors are positive (SAp is proven to under-approximate the exact MTTF)

m ↑ \ k →	2	3	4	5	6	7	8	9	10	11	12
11											81.45
10										79.25	69.20
9									76.61	65.85	58.25
8								73.38	62.01	54.73	47.79
7							69.36	57.75	50.31	44.30	39.06
6						64.29	52.99	44.91	40.11	35.35	31.43
5					57.84	47.34	39.12	34.28	31.22	28.13	25.06
4				49.62	39.96	33.44	28.17	24.82	22.98	21.64	20.28
3			39.29	30.21	25.73	22.68	20.09	17.80	15.93	14.55	13.63
2		25.91	19.44	15.77	13.60	12.30	11.50	10.98	10.62	10.35	10.13
1	05.76	05.76	05.76	05.76	05.76	05.76	05.76	05.76	05.76	05.76	05.76

- Error > 50 %
- 25 % < Error ≤ 50 %
- Error ≤ 25 %

SAp is reasonably accurate

Example: If $MTTF_{exact} = 10^9$ hours, 100% error ➡ $MTTF_{SAp} = 0.5 \times 10^9$ hours

Relative errors significant even for small k

➡ Exact analysis needed when feasible

Summary

Approach	Accuracy	Expressiveness	Scalability
PMC	Exact	General system, any weakly-hard constraint	Poor ($m \leq 11$)
Mart	Exact	IID systems, any weakly-hard constraint	Poor ($m \leq 16$)
SAp	Sound approx. ($\leq$ 100%)	IID systems, single (m, k) constraint	Good ($m \leq 1000$)

Summary

Approach	Accuracy	Expressiveness	Scalability
PMC	Exact	General system, any weakly-hard constraint	Poor ($m \leq 11$)
Mart	Exact	IID systems, any weakly-hard constraint	Poor ($m \leq 16$)
SAp	Sound approx. ($\leq$ 100%)	IID systems, single (m, k) constraint	Good ($m \leq 1000$)

More in the paper!

- ➡ PRISM code and Mart example
- ➡ PMC / Mart for $\langle m, k \rangle$ and $\overline{\langle m \rangle}$ constraints
- ➡ SAp details and soundness proofs
- ➡ More extensive evaluation of PRISM

Summary

Approach	Accuracy	Expressiveness	Scalability
PMC	Exact	General system, any weakly-hard constraint	Poor ($m \leq 11$)
Mart	Exact	IID systems, any weakly-hard constraint	Poor ($m \leq 16$)
SAp	Sound approx. ($\leq$ 100%)	IID systems, single (m, k) constraint	Good ($m \leq 1000$)

Future work: Make **SAp more expressive**

- ➡ Handle other / multiple weakly-hard constraints
- ➡ Beyond IID iteration failure probabilities

More in the paper!

- ➡ PRISM code and Mart example
- ➡ PMC / Mart for $\langle m, k \rangle$ and $\langle \overline{m} \rangle$ constraints
- ➡ SAp details and soundness proofs
- ➡ More extensive evaluation of PRISM