Measuring Masking Fault-Tolerance in Stochastic Systems

Pablo Castro, Pedro D’Argenio, Luciano Putruele, Ramiro Demasi

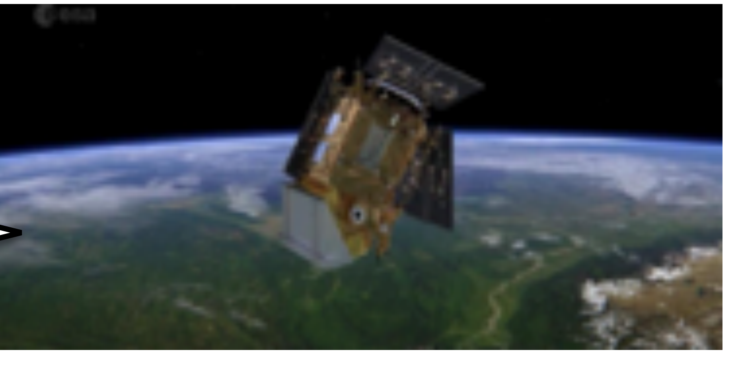
Motivation

Fault-Tolerance can be defined as the capability of systems to continue operating in a correct way even under the occurrence of faults

Mobile phones

Avionics software

Cryptocurrencies



Satellites

Nominal Models and Fault Model

Nominal Model

What is the relation?

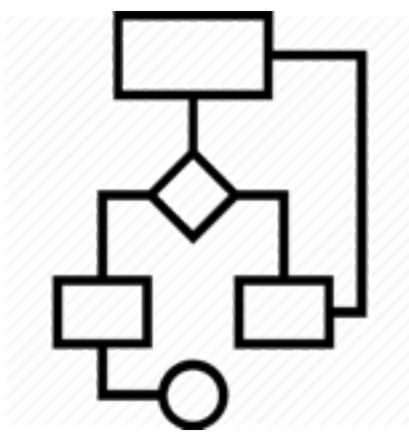
Implementation

A description of the system
In which faults are not taken into accounts

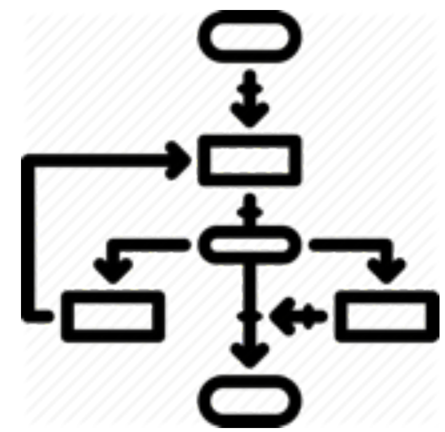
Nominal System
+Faults
+Fault-tolerant mechanisms

Many Possible Fault-Tolerant Implementations

Implementation 2



Implementation 1



Implementation 3

Which implementation provides more fault-tolerance?

Hard to say in practice

Classifying Fault-Tolerance

We can classify fault-tolerance taking into account the kind of properties preserved by the system after the occurrence of faults:

- **Liveness properties**: Something good eventually happens.
- **Safety properties**: nothing bad happens.

Lineal Temporal Properties can be described as a combination of both

Failsafe Fault-Tolerance

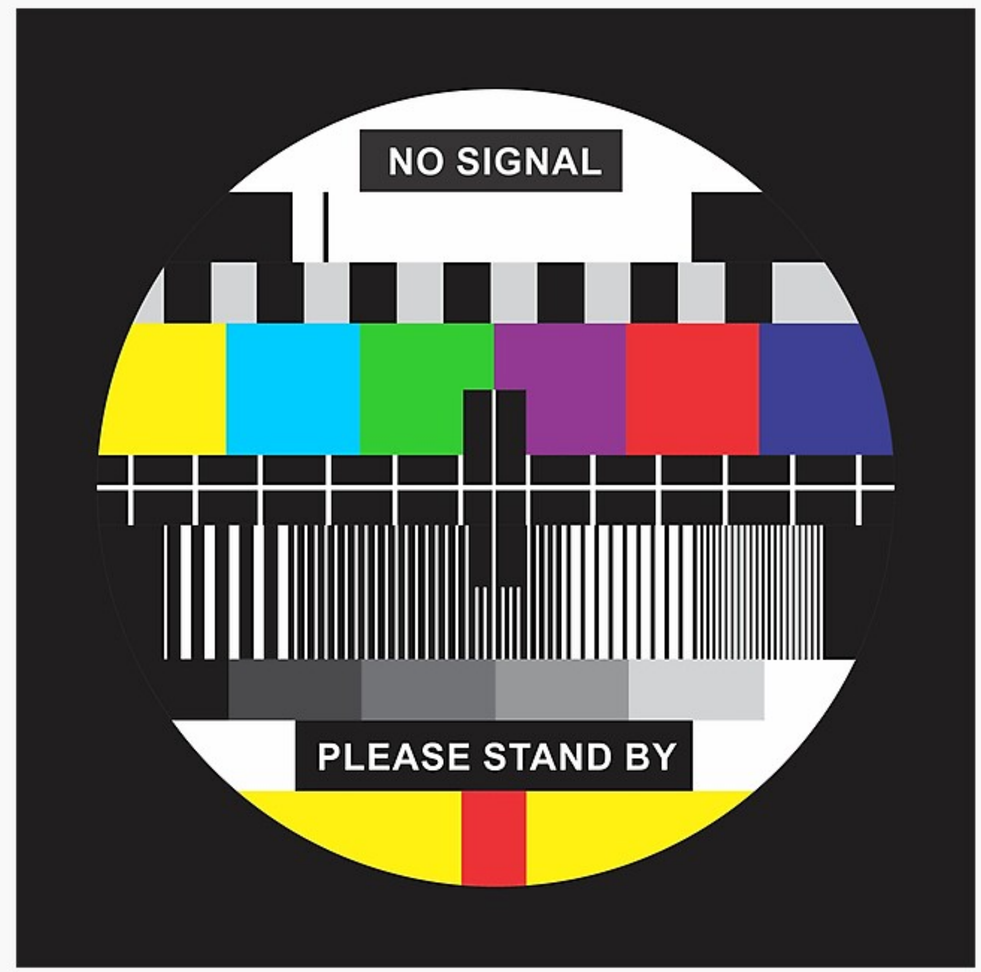
- The system is taken to a safe state after the occurrence of faults.
- Important in systems in which preserving safety properties is more relevant than progress
- Simple example: Any elevator system.

Non-masking Fault-Tolerance

- The system may show an incorrect behavior after a fault, but eventually it recovers the correct behavior.

- Liveness properties are preserved

- Simple example: streaming platforms.



Masking Fault-Tolerance

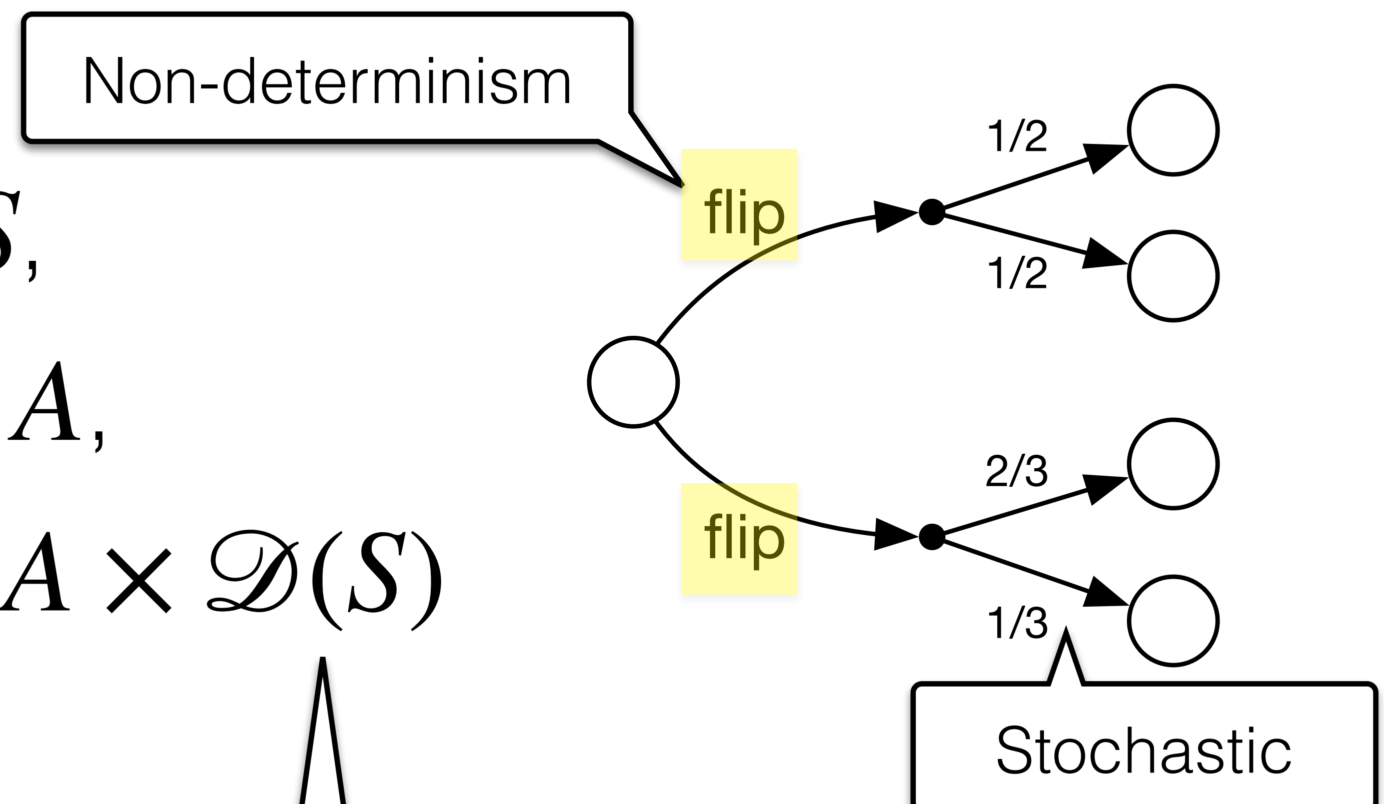
- The occurrence of faults are not visible for the users.
- Safety+Liveness properties preserved
- Examples of masking fault-tolerance are systems that use some kind of **redundancy**.

We only will focus on this kind of fault-tolerance

Probabilistic Models

We use Probabilistic Transition Systems (PTSs) to model probabilistic systems/protocols/software.

- A finite set of states S,
- A finite set of actions A,
- A relation $\rightarrow \subseteq S \times A \times \mathscr{D}(S)$

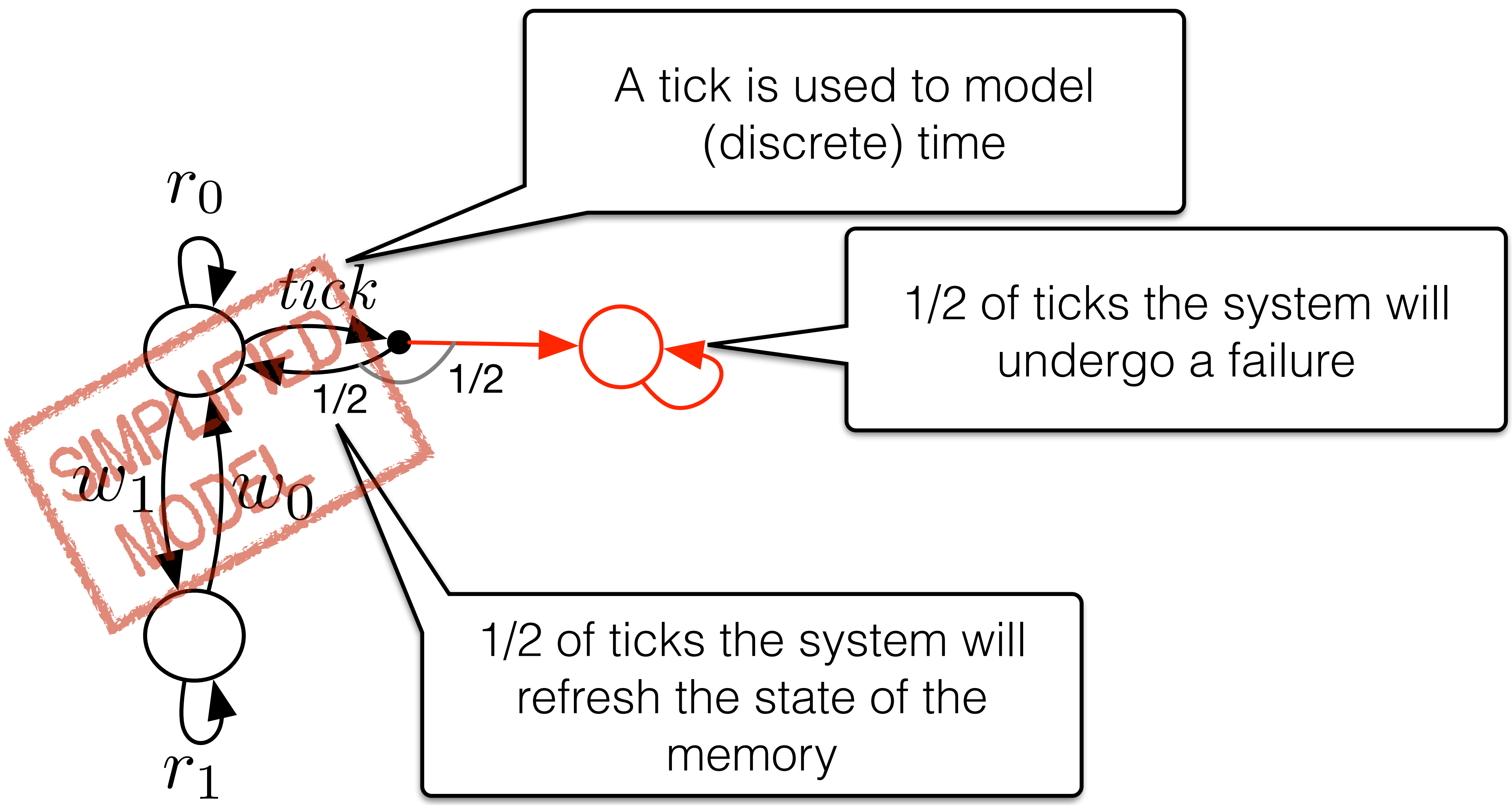


This admits both stochastic behavior and non-determinism

$\mathscr{D}(S)$ is the set of distributions $\mu : S \rightarrow [0,1]$

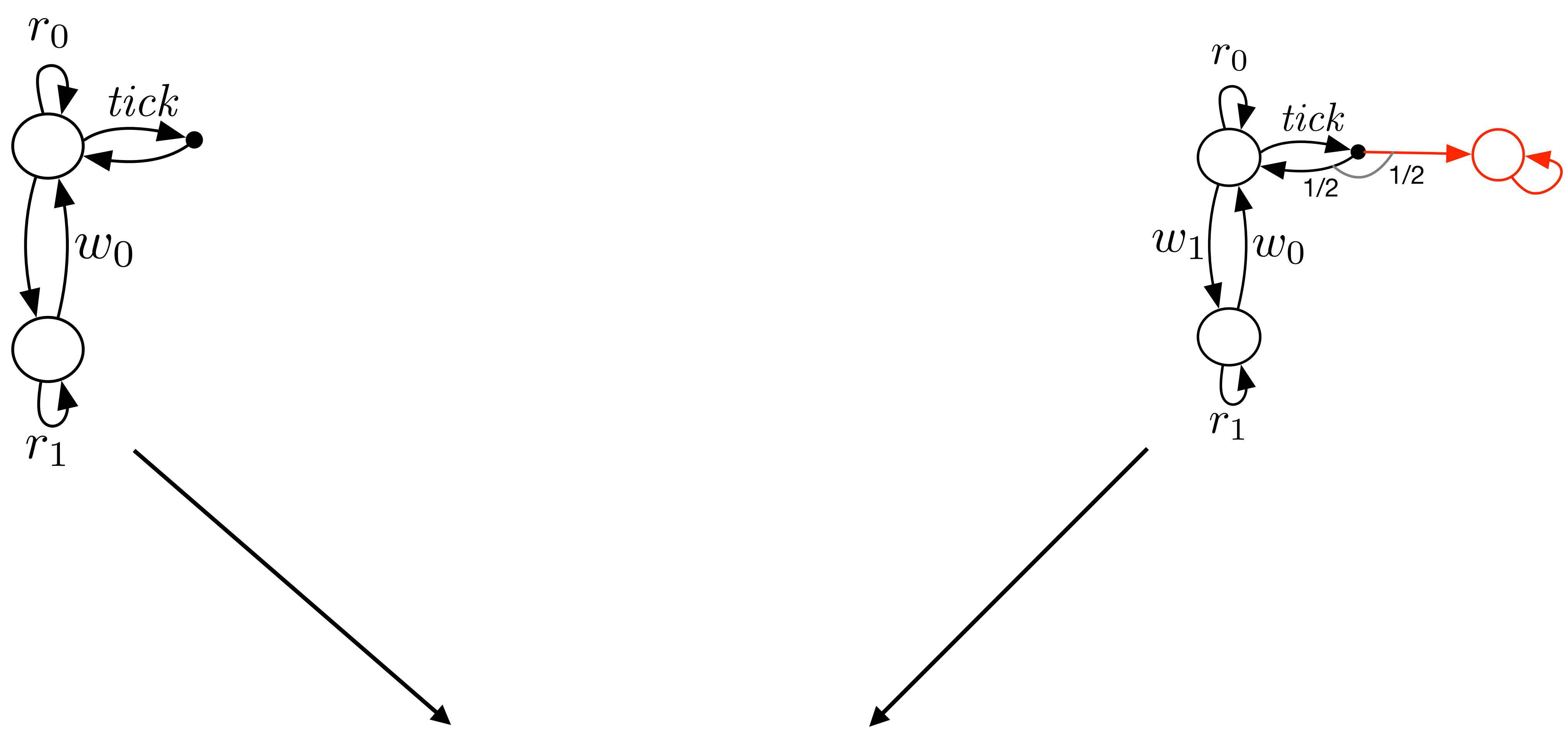
Modeling Faults

We use probabilities to introduce the possibility of the occurrence of faults



Idea

Two models:



We play a game between Verifier and Refuter

Verifier: tries to prove the system is fault -tolerant

Refuter: tries to disprove this

Couplings

To be able of modeling simulation relations we need couplings.

Given $\mu : S \to [0,1]$ and $\mu' : S' \to [0,1]$ $w : S \times S' \to [0,1]$

is a **coupling** if: $w(S, -) = \mu'$ and $w(-, S') = \mu$

Couplings can be defined as the solutions of some linear (in)equalities:

$$\sum_{s_j \in supp(\mu')} x_{s_k,s_j} = \mu(s_k), \text{ for } s_k \in supp(\mu)$$

$$\sum_{s_k \in supp(\mu)} x_{s_k,s_j} = \mu'(s_j), \text{ for } s_j \in supp(\mu')$$

$$x_{s_j,s_k} \geq 0, \text{ for } s_k \in supp(\mu) \text{ and } s_j \in supp(\mu')$$

A Masking Probabilistic Game

We define a stochastic two-player game:

Two players:

- The Refuter
- The Verifier

These games allow one to capture probabilistic bisimulation relation as well as devise quantitative extensions of it.

The Game

Given two PTSs A and A' we define a game $G_{A,A'}$

- The **Refuter** starts selecting some $s \xrightarrow{a} \mu$ or $s' \xrightarrow{a} \mu'$,
- The **Verifier** tries to mimic the action, selects $s' \xrightarrow{a} \mu'$ and a coupling $w : S \times S' \to [0,1]$ for μ and μ'
- If the **Refuter** chose a fault ($s \xrightarrow{f} \mu'$) the **Verifier** must chose Δ_s (Dirac distribution)
- Then, the game moves in a randomized way following the coupling.

Formal Definition

States are nodes of the type:

$$(s, a, s', \mu, \mu', w)$$

Refuter's state

Last action played

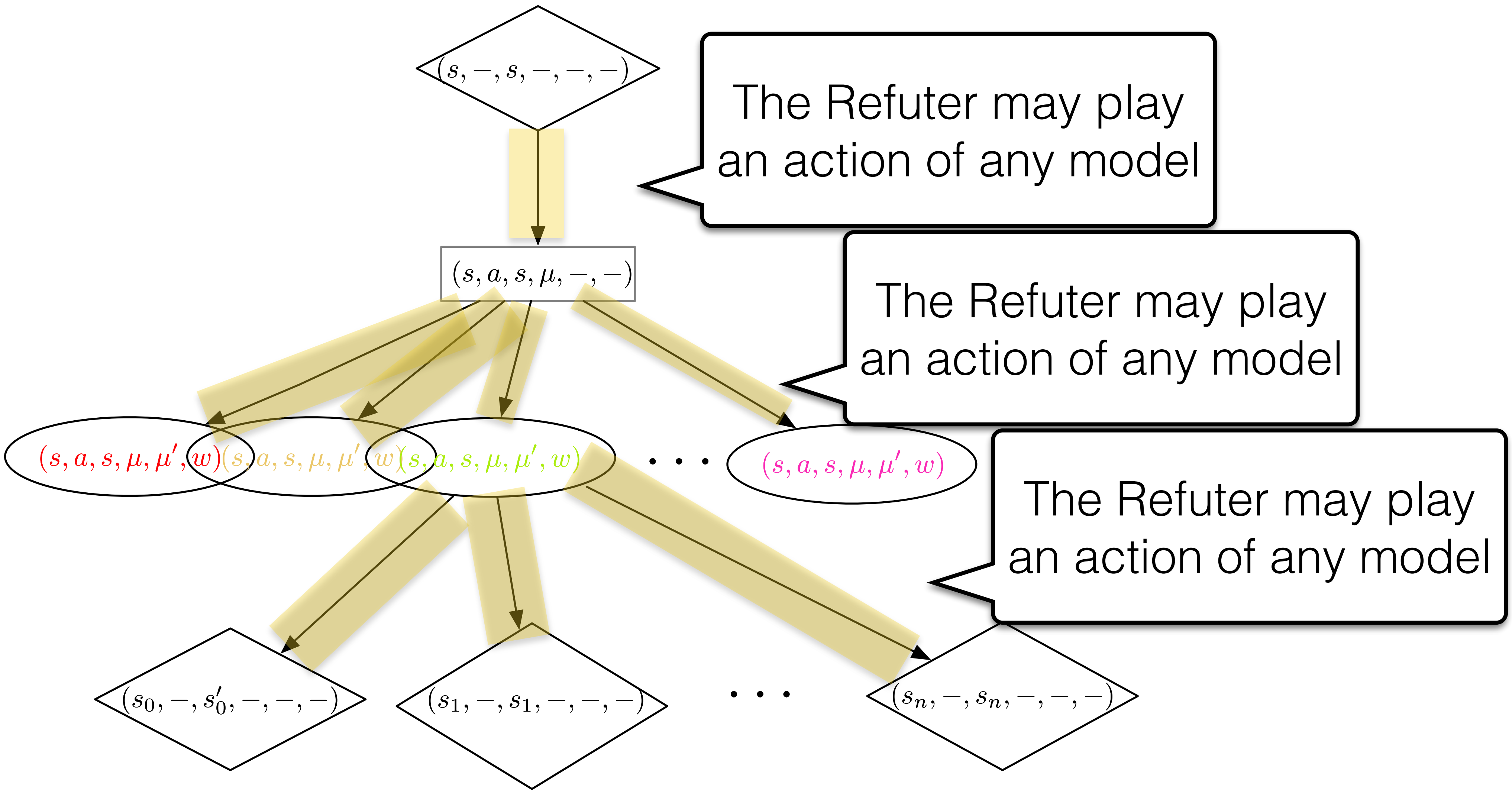
Verifier's state

Verifier's distribution

Refuter's distribution

Coupling selected

Formal definitions of plays



Boolean Game Objective and Results

When there are no faults, this captures probabilistic bisimulation

- The Refuter wins if the error state is reached,
- The Verifier wins if the error state is never reached

Both players has optimal memoryless strategies

The value can be computed in polynomial time

Symbolic Games

Recall: couplings can be described by means of equations:

$$\sum_{s_j \in supp(\mu')} x_{s_k,s_j} = \mu(s_k), \text{ for } s_k \in supp(\mu)$$

$$\sum_{s_k \in supp(\mu)} x_{s_k,s_j} = \mu'(s_j), \text{ for } s_j \in supp(\mu')$$

Describes all the couplings between μ and μ'

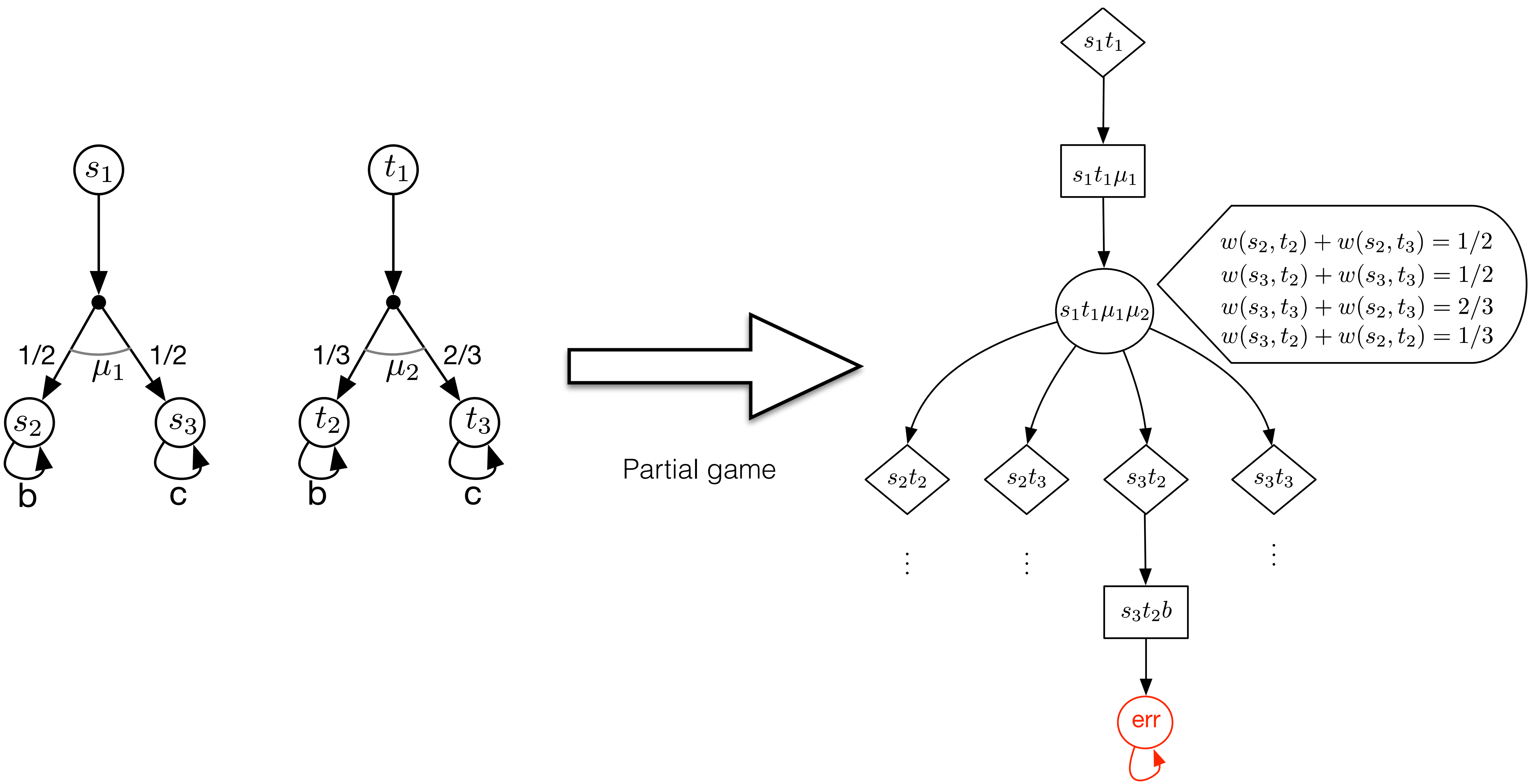
$$x_{s_j,s_k} \geq 0, \text{ for } s_k \in supp(\mu) \text{ and } s_j \in supp(\mu')$$

Instead of explicitly adding couplings, we decorate games with equations:

Example

Two (non-bisimilar) PTSs

Corresponding symbolic game



Using the Symbolic Game

We can use the symbolic game to solve the game

$$
\begin{aligned}
U^0 &= \{v_{err}\},\\
U^{i+1} &= \{v' \mid v' \in V^{\mathcal{SG}}_{\mathsf{R}} \wedge Post(v') \cap U^i \neq \emptyset\} \cup\\
&\quad \{v' \mid v' \in V^{\mathcal{SG}}_{\mathsf{V}} \wedge Post(v') \subseteq \textstyle\bigcup_{j \leq i} U^j \wedge Post(v') \cap U^i \neq \emptyset\} \cup\\
&\quad \{v' \mid v' \in V^{\mathcal{SG}}_{\mathsf{P}} \wedge Post(v') \cap U^i \neq \emptyset \wedge Eq(v', Post(v') \cap U^i) \textit{ has no solution}\}
\end{aligned}
$$

These sets capture vertices from which the Refuter has winning plays

$$\sum_{s_j \in supp(\mu')} x_{s_k, s_j} = \mu(s_k), \text{ for } s_k \in supp(\mu)$$

$$\sum_{s_k \in supp(\mu)} x_{s_k, s_j} = \mu'(s_j), \text{ for } s_j \in supp(\mu')$$

$$x_{s_j, s_k} \geq 0, \text{ for } s_k \in supp(\mu) \text{ and } s_j \in supp(\mu')$$

$$\sum_{(s,-,s',-,-) \in Post(v') \cap U^i} x_{s,s'} = 0$$

For $v = (s, -, s', \mu, \mu')$

No coupling with probability 0 of going to U^i

Quantitative Games

Instead of saying if there is a masking (bi)simulation or not, we can consider a quantitative objetive

- We consider some actions $M \subseteq Act$ as being milestone to count,
- A reward is defined as: $r(v) = v[1] \in M ? 1 : 0$
- Then we define a function: $f_m(v_0 v_1 v_2 \ldots) = \sum_{i=0}^{\infty} r(v_i)$

well-defined in reals when the game stops

The Verifier tries to maximize the expected value of f_m, and the Refuter tries to minimizes it.

Stopping Conditions

The objective of the game is to maximize/minimize:

$$\mathbb{E}^{\pi_V,\pi_R}_{\mathcal{G},v_0^{\mathcal{G}}}[f_m] = \lim_{n\to\infty} \mathbb{E}^{\pi_V,\pi_R}_{\mathcal{G},v_0^{\mathcal{G}}}[\lambda\rho \,.\, \sum_{i=0}^{n} r^{\mathcal{G}}_{\mathrm{m}}(\rho_i)]$$

Where:

π_V is the strategy played by the Verifier

π_R is the strategy played by the Refuter

Standard stopping condition: $Prob^{\pi_V,\pi_R}_{\mathcal{G},v}(\Diamond v_{error}) = 1$

For every pair of memoryless strategies

That is: a terminal state will be reached with probability 1

A More General Condition

Consider the following:

```
module NOMINAL
 b :  [0..1] init 0;
 m :  [0..1] init 0; // 0 = normal,
                     // 1 = refreshing

 [w0]   (m=0)         -> (b'= 0);
 [w1]   (m=0)         -> (b'= 1);
 [r0]   (m=0) & (b=0) -> true;
 [r1]   (m=0) & (b=1) -> true;
 [tick] (m=0)         -> p:  (m'= 1) +
                         (1-p):  true;

 [rfsh] (m=1)         -> (m'= 0);
endmodule
```

```
module FAULTY
 v :  [0..3] init 0;
 s :  [0..2] init 0;   // 0 = normal, 1 = faulty,
                       // 2 = refreshing
 [w0]    (s!=2)            -> (v'= 0) & (s'= 0);
 [w1]    (s!=2)            -> (v'= 3) & (s'= 0);
 [r0]    (s!=2) & (v<=1)  -> true;
 [r1]    (s!=2) & (v>=2)  -> true;
 [tick]  (s!=2)            -> p:  (s'= 2) + q:  (s'= 1)
                              + (1-p-q): true;
 [rfsh]  (s=2)             -> (s'=0)
                              & (v'= (v<=1) ? 0 : 3);
 [fault] (s=1)             -> (v'= (v<3) ? (v+1) : 2)
                              & (s'= 0);
 [fault] (s=1)             -> (v'= (v>0) ? (v-1) : 1)
                              & (s'= 0) ;

endmodule
```

Reading

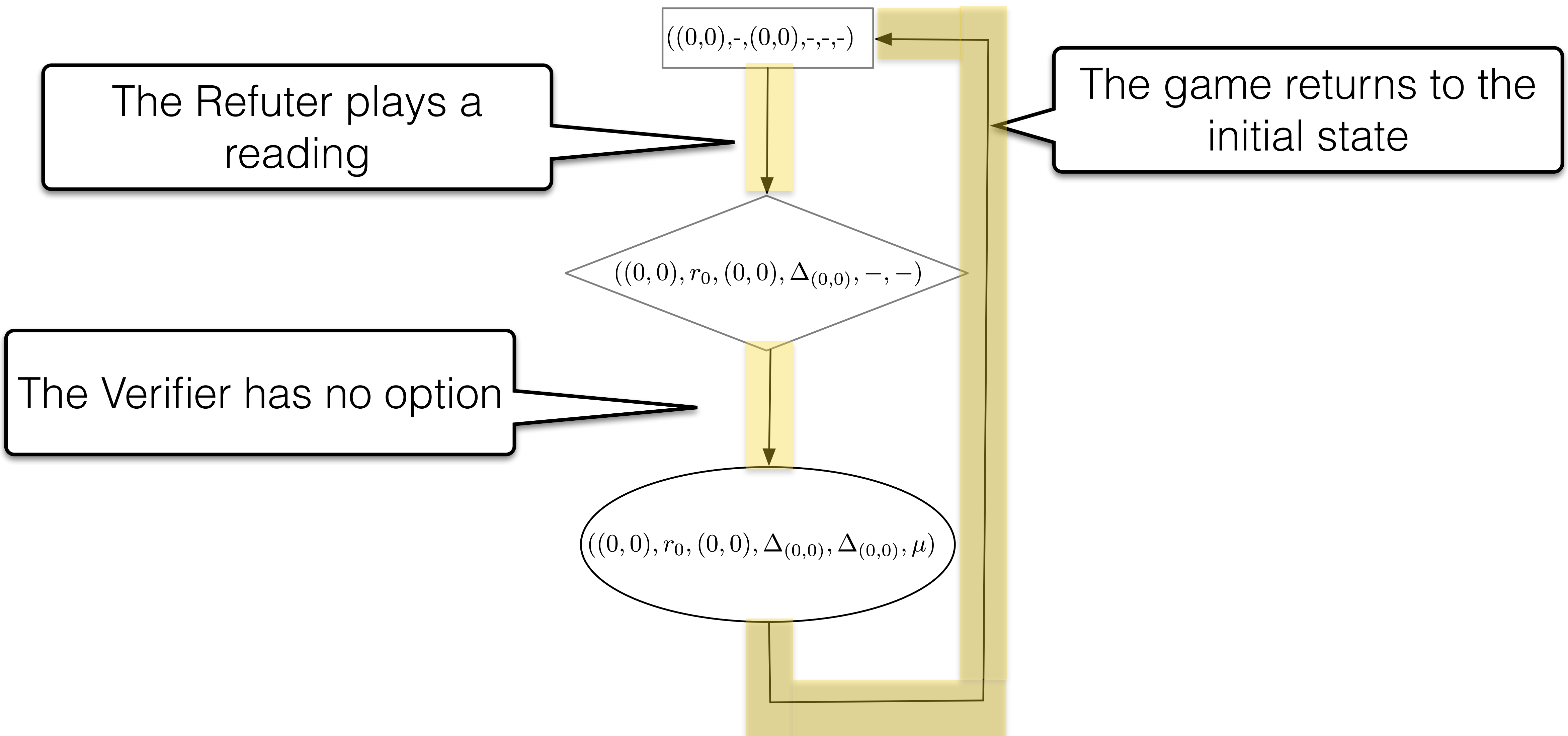
Writing

Fault

Two versions of the memory model

Let's play the game

Consider the following play:



The error state is never reached, the Refuter plays in such a way that it keeps the system away from failing!

Fair Plays

For avoiding this kind of behavior from the Refuter, we assume that she behaves in a fair way:

The set of fair play is defined as:

$$RFP = \{\rho \in \Omega \mid v \in inf(\rho) \cap V_R \Rightarrow Post(v) \subseteq inf(\rho)\}$$

A strategy π_R for the refuter is said to be a.s. fair if:

$$\mathbb{P}^{\pi_V,\pi_R}_{\mathcal{G},v}(RFP) = 1 \qquad \text{For all } \pi_V$$

We are interested in games that stop under fairness:

For π_R fair: $$\mathbb{P}^{\pi_V,\pi_R}_{\mathcal{G},v}(\Diamond v_{err}) = 1$$

The game ends with probability one

Some questions

- **Q1** Are the value of these games well-defined in $\mathbb{R}$? Furthermore, Are they determined?

- **Q2** How can we compute the values of these infinite games?

- **Q3** Can we use the symbolic games to compute the value?

Defining a subgame

For answering that questions we consider a subgame:

Given two distributions: μ, μ'

$$\sum_{s_j \in supp(\mu')} x_{s_k,s_j} = \mu(s_k), \text{ for } s_k \in supp(\mu)$$

$$\sum_{s_k \in supp(\mu)} x_{s_k,s_j} = \mu'(s_j), \text{ for } s_j \in supp(\mu')$$

$$x_{s_j,s_k} \geq 0, \text{ for } s_k \in supp(\mu) \text{ and } s_j \in supp(\mu')$$

Defines a polytope

Finite, but an exponential number of vertices

The game $\mathscr{H}_{A,A'}$ has the same maximizer and minimizer vertices as $\mathscr{G}_{A,A'}$ but their probabilistic vertices are the vertices of the polytope

Results

We can prove that the infinite game is determined using de restricted game:

If $\mathscr{H}_{A,A'}$ is stopping under fairness then:

Follows from property of finite games: CAV 22

$$\inf_{\pi_{\mathsf{R}} \in \Pi^{f}_{\mathsf{R},\mathcal{G}}} \sup_{\pi_{\mathsf{V}} \in \Pi_{\mathsf{V},\mathcal{G}}} \mathbb{E}^{\pi_{\mathsf{V}},\pi_{\mathsf{R}}}_{\mathcal{G},v}[f_{\mathsf{m}}] = \inf_{\pi_{\mathsf{R}} \in \Pi^{MDf}_{\mathsf{R},\mathcal{H}}} \sup_{\pi_{\mathsf{V}} \in \Pi^{MD}_{\mathsf{V},\mathcal{H}}} \mathbb{E}^{\pi_{\mathsf{V}},\pi_{\mathsf{R}}}_{\mathcal{H},v}[f_{\mathsf{m}}]$$

$$= \sup_{\pi_{\mathsf{V}} \in \Pi^{MD}_{\mathsf{V},\mathcal{H}}} \inf_{\pi_{\mathsf{R}} \in \Pi^{MDf}_{\mathsf{R},\mathcal{H}}} \mathbb{E}^{\pi_{\mathsf{V}},\pi_{\mathsf{R}}}_{\mathcal{H},v}[f_{\mathsf{m}}] = \sup_{\pi_{\mathsf{V}} \in \Pi_{\mathsf{V},\mathcal{G}}} \inf_{\pi_{\mathsf{R}} \in \Pi^{f}_{\mathsf{R},\mathcal{G}}} \mathbb{E}^{\pi_{\mathsf{V}},\pi_{\mathsf{R}}}_{\mathcal{G},v}[f_{\mathsf{m}}].$$

The next problem is: how can we compute the game value?

Solving the Game

We can solve the game using Bellman equations over the symbolic game.

If $\mathscr{H}_{A,A'}$ is stopping under fairness then,

the value of the game is gfp of:

Vertices of the polytope

$$\Gamma(f)(v) = \begin{cases} \min\left(\mathbf{U}, \max_{w \in \mathbb{V}(\mathbb{C}(v[3],v[4]))} \sum_{v' \in Post(v)} w(v'[0], v'[2]) f(v')\right) & \textit{if } v \in V_{\mathsf{P}}^{\mathcal{SG}} \\ \min\left(\mathbf{U}, r_{\mathsf{m}}^{\mathcal{SG}}(v) + \max\left\{f(v') \mid v' \in Post(v)\right\}\right) & \textit{if } v \in V_{\mathsf{V}}^{\mathcal{SG}} \\ \min\left(\mathbf{U}, \min\left\{f(v') \mid v' \in Post(v)\right)\right\} & \textit{if } v \in V_{\mathsf{R}}^{\mathcal{SG}} \setminus \{v_{err}\} \\ 0 & \textit{if } v = v_{err} \end{cases}$$

Open Questions

We can prove that the game is determined, but:

- If the restricted game stops under fairness with prob. 1, then the infinite game stops with probability one?

- When one add negative numbers, there could not be optimal memoryless strategies, or the game may have not a value. **What conditions are needed for guaranteeing this**?