Quantifying Masking Fault-Tolerance via Fair Stochastic Games

Pablo F. Castro, Pedro R. D'Argenio, Ramiro Demasi, Luciano Putruele

UN Córdoba - UN Río Cuarto - CONICET

http://www.cs.famaf.unc.edu.ar/~dargenio/

CONICET

EXPRESS/SOS 2023

UNC

Motivation: a memory cell

```
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
```

Motivation: a memory ce

Implementation model:
Includes faults and fault handling mechanism

```
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
```

Nominal model:
Prescribes the normal behavior where faults do not occur

Motivation: a memory ce

Implementation model:
Includes faults and fault handling mechanism

```
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
```

Nominal model:
Prescribes the normal behavior where faults do not occur

```
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
```

CO

How "close" is the implementation model to the nominal model?

...ion: a memory ce...

Implementation model: Includes faults and fault handling mechanism

```
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
```

Nominal model: Prescribes the normal behavior where faults do not occur

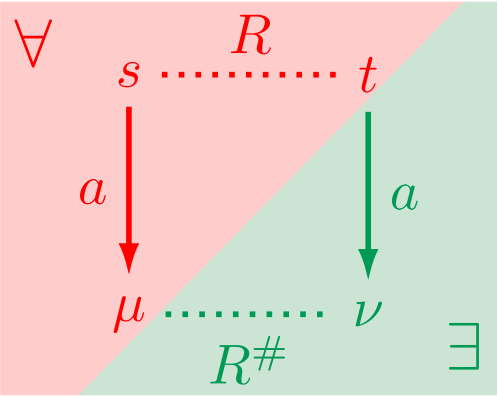
We want a general technique in contraposition with *ad-hoc* techniques

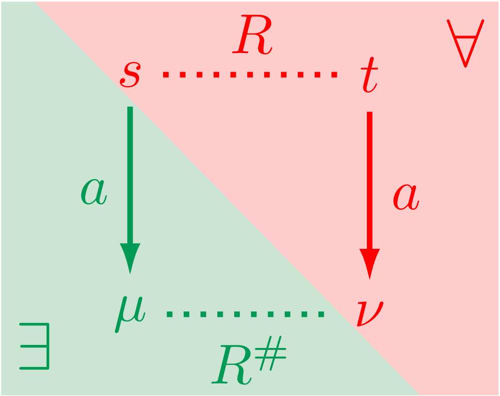
CO...

UNC

Probabilistic Masking Simulation: when implementations are perfect

For non-faulty actions





So far, this is the usual probabilistic bisimulation

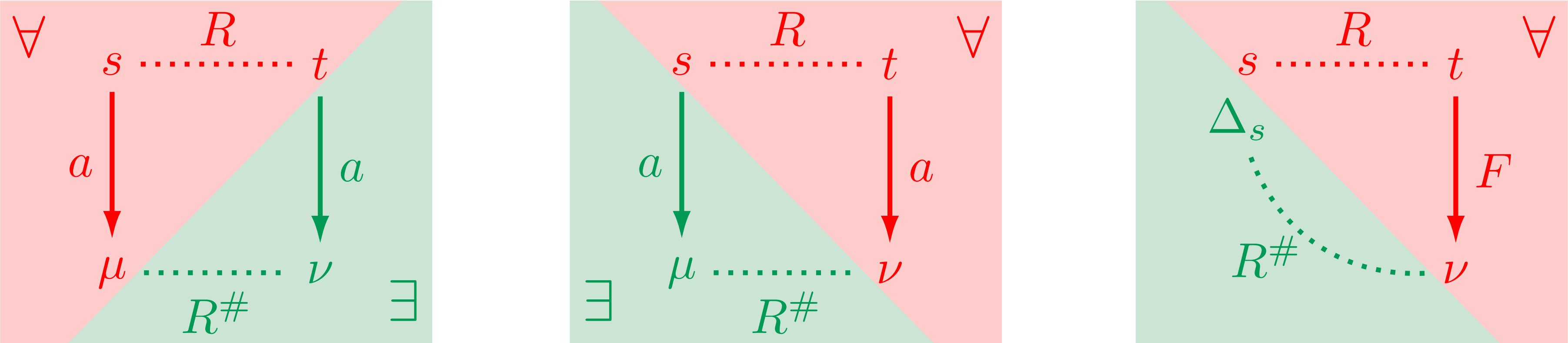
exists a coupling w s.t.
$w(s', t') > 0 \;\Rightarrow\; (s', t') \in R$

a distribution in $S \times S$ s.t.
$w(\cdot, S) = \mu$ and $w(S, \cdot) = \nu$

Probabilistic Masking Simulation: when implementations are perfect

For non-faulty actions

where F is a fault



exists a coupling w s.t.
$w(s',t') > 0 \;\Rightarrow\; (s',t') \in R$

a distribution in $S \times S$ s.t.
$w(\cdot, S) = \mu$ and $w(S, \cdot) = \nu$

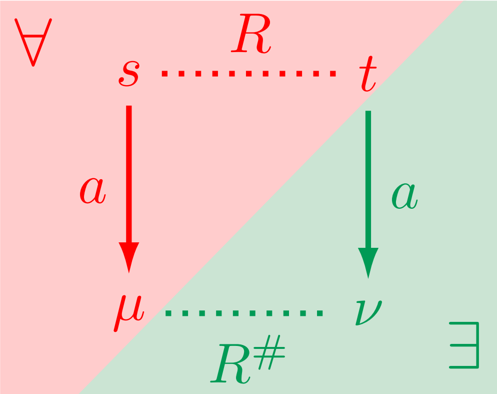
CONICET

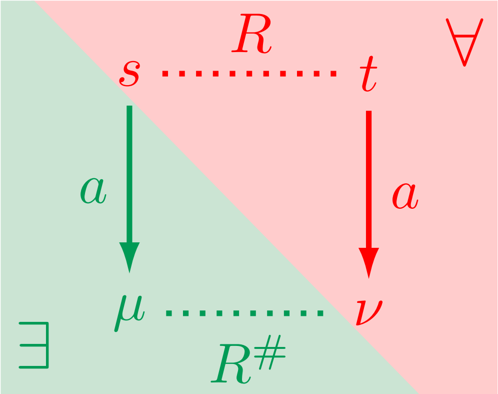
UNC

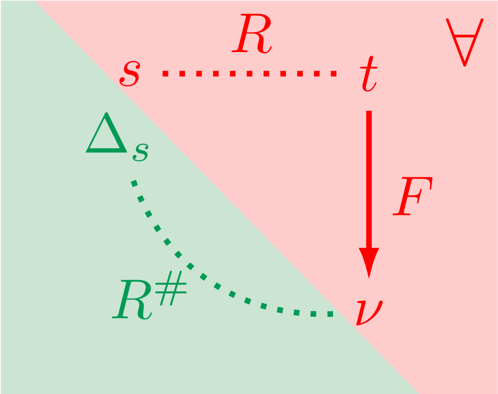
Probabilistic Masking Simulation: when implementations are perfect

For non-faulty actions

where F is a fault







exists a coupling w s.t.
$w(s',t') > 0 \;\Rightarrow\; (s',t') \in R$

a distribution in $S \times S$ s.t.
$w(\cdot, S) = \mu$ and $w(S, \cdot) = \nu$

The set $\mathbb{C}(\mu,\nu)$ of all couplings forms a polytope with vertices $\mathbb{V}(\mathbb{C}(\mu,\nu))$

CONICET

UNC

Probabilistic Masking Simulation: when implementations are perfect

```
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
```

$\preceq_m$

```
module FAULTY

 v : [0..3] init 0;
 s : [0..2] init 0;   // 0 = normal, 1 = faulty,
                      // 2 = refreshing
 f : [0..1] init 0;   // fault limiting artifact

 [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) & (f<1)    -> (v'= (v<3) ? (v+1) : 2)
                              & (s'= 0) & (f'= f+1);
 [fault]  (s=1) & (f<1)    -> (v'= (v>0) ? (v-1) : 1)
                              & (s'= 0) & (f'= f+1);

endmodule
```

$$R = \{\langle (\mathtt{b},\mathtt{m}), (\mathtt{v},\mathtt{s},\mathtt{f})\rangle \mid 2\mathtt{b} \leq \mathtt{v} \leq 2\mathtt{b}{+}1 \wedge (\mathtt{m} = 1 \Leftrightarrow \mathtt{s} = 2)\}$$

Probabilistic Masking Simulation: when implementations are perfect

```
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
```

$\not\preceq_m$

```
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
```

A characterizing stochastic game

1. The refuter R chooses either $\begin{cases} s \xrightarrow{a} \mu \text{ from the nominal model or} \\ s' \xrightarrow{a}{}' \mu' \text{ from the implementation;} \end{cases}$

2a. If $a \notin \mathcal{F}$, the verifier V chooses $\begin{cases} s' \xrightarrow{a}{}' \mu' \text{ if R chose from the nominal, or} \\ s \xrightarrow{a} \mu \text{ otherwise.} \end{cases}$

In addition, V chooses a coupling w for $\mathbb{C}(\mu, \mu')$;

2b. If $a \in \mathcal{F}$, V can only select Δ_s and the only coupling $w \in \mathbb{C}(\Delta_s, \mu')$;

3. The successor pair of states (t, t') is chosen probabilistically according to w.

If the play continues forever, V wins and there is a probabilistic masking simulation

CONICET

A characterizing stochastic game

Stochastic masking game graph

1. The refuter R chooses either $\begin{cases} s \xrightarrow{a} \mu \text{ from the nominal model or} \\ s' \xrightarrow{a}{}' \mu' \text{ from the implementation;} \end{cases}$

2a. If $a \notin \mathcal{F}$, the verifier V chooses $\begin{cases} s' \xrightarrow{a}{}' \mu' \text{ if R chose from the nominal, or} \\ s \xrightarrow{a} \mu \text{ otherwise.} \end{cases}$

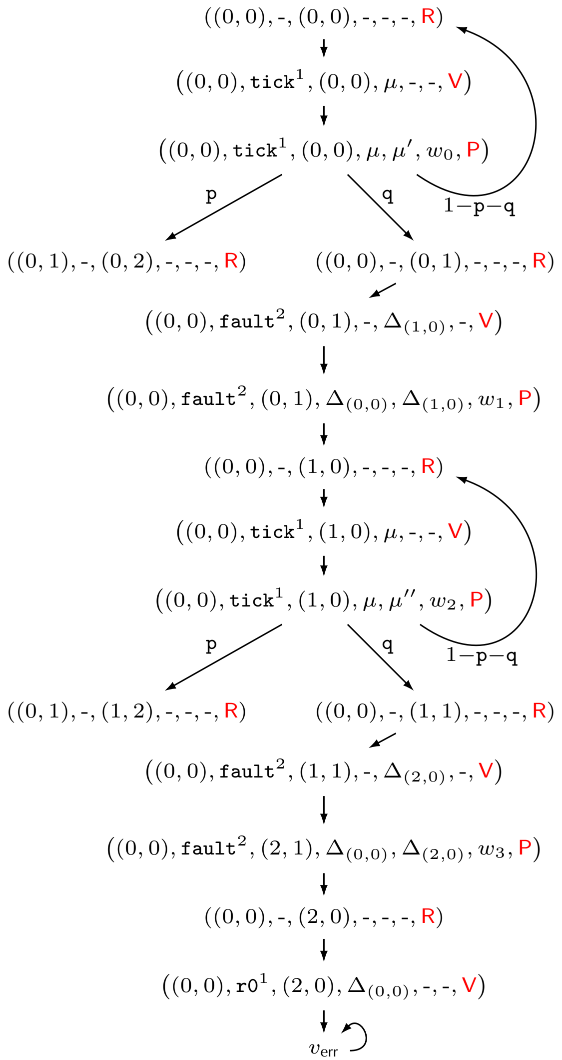
In addition, V chooses a coupling w for $\mathbb{C}(\mu, \mu')$;

2b. If $a \in \mathcal{F}$, V can only select Δ_s and the only coupling $w \in \mathbb{C}(\Delta_s, \mu')$;

3. The successor pair of states (t, t') is chosen probabilistically according to w.

```
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
```



CONICET

UNC

A characterizing stochastic game

Stochastic masking game graph

1. The refuter R chooses either $\begin{cases} s \xrightarrow{a} \mu \text{ from the nominal model or} \\ s' \xrightarrow{a}{}' \mu' \text{ from the implementation;} \end{cases}$

2a. If $a \notin \mathcal{F}$, the verifier V chooses $\begin{cases} s' \xrightarrow{a}{}' \mu' \text{ if R chose from the nominal, or} \\ s \xrightarrow{a} \mu \text{ otherwise.} \end{cases}$

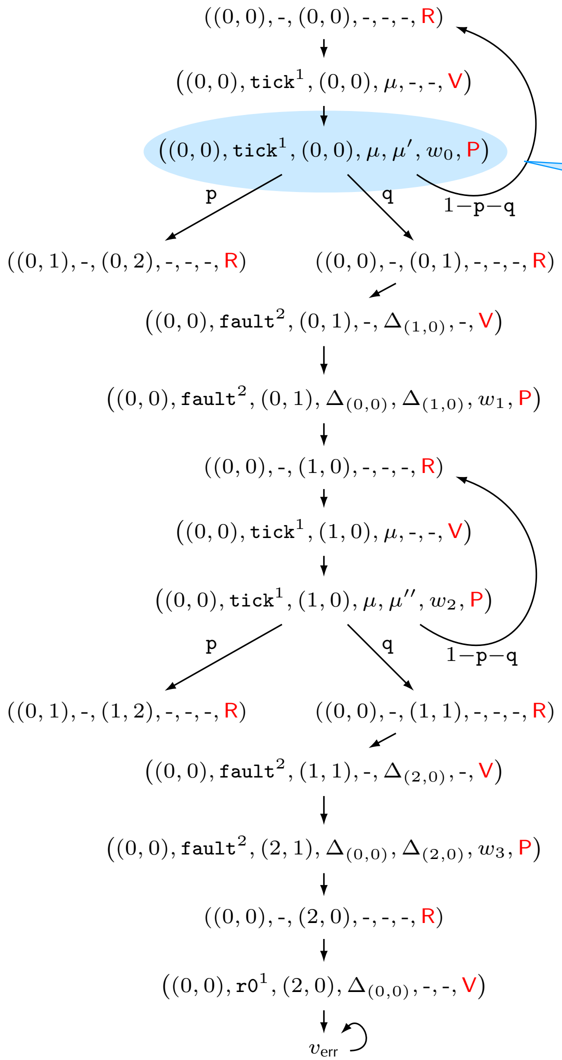
In addition, V chooses a coupling w for $\mathbb{C}(\mu, \mu')$;

2b. If $a \in \mathcal{F}$, V can only select Δ_s and the only coupling $w \in \mathbb{C}(\Delta_s, \mu')$;

3. The successor pair of states (t, t') is chosen probabilistically according to w.

```
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
```



Uncountable branching since $w_0 \in \mathbb{C}(\mu, \mu')$

CONICET

UNC

A characterizing stochastic game

1. The refuter R chooses either $\begin{cases} s \xrightarrow{a} \mu \text{ from the nominal model or} \\ s' \xrightarrow{a}{}' \mu' \text{ from the implementation;} \end{cases}$

2a. If $a \notin \mathcal{F}$, the verifier V chooses $\begin{cases} s' \xrightarrow{a}{}' \mu' \text{ if R chose from the nominal, or} \\ s \xrightarrow{a} \mu \text{ otherwise.} \end{cases}$

In addition, V chooses a coupling w for $\mathbb{C}(\mu, \mu')$;

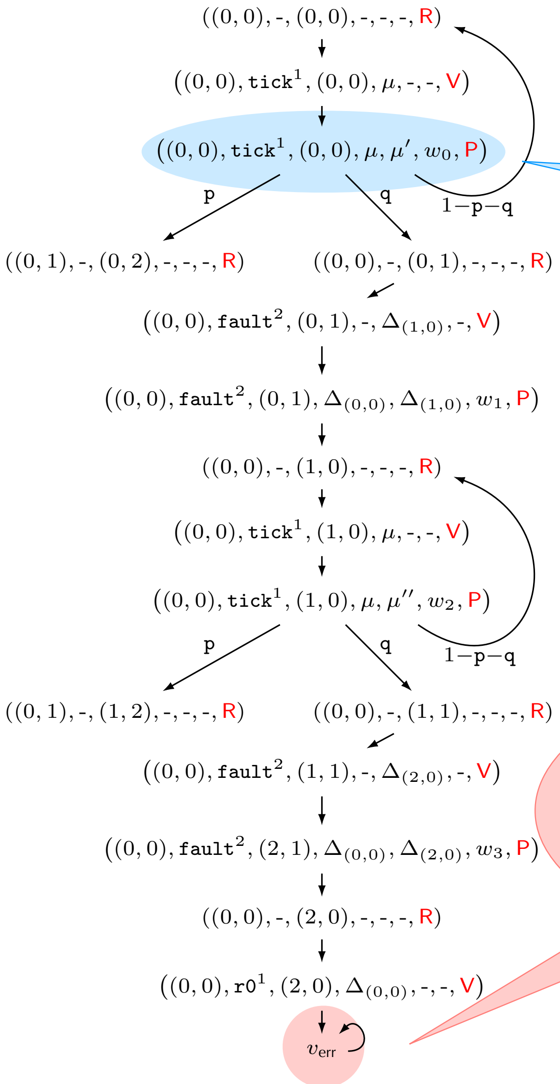
2b. If $a \in \mathcal{F}$, V can only select Δ_s and the only coupling $w \in \mathbb{C}(\Delta_s, \mu')$;

3. The successor pair of states (t, t') is chosen probabilistically according to w.

```
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
```

Stochastic masking game graph



Uncountable branching since $w_0 \in \mathbb{C}(\mu, \mu')$

Probabilistic masking simulation iff

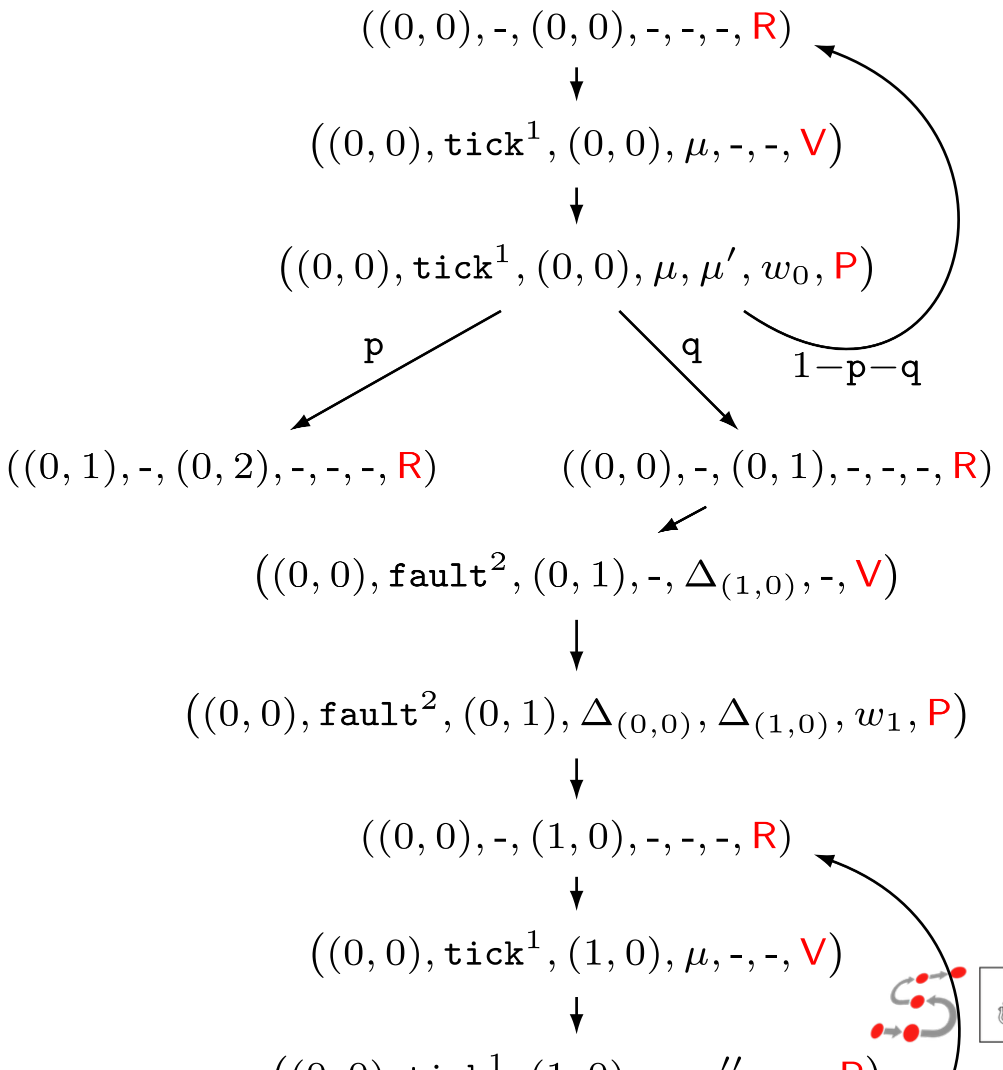
$$\sup_{\pi_R \in \Pi_R} \inf_{\pi_V \in \Pi_V} Prob^{\pi_V, \pi_R}(\Diamond\, v_{err}) = 0$$

CONICET

UNC

A characterizing stochastic game

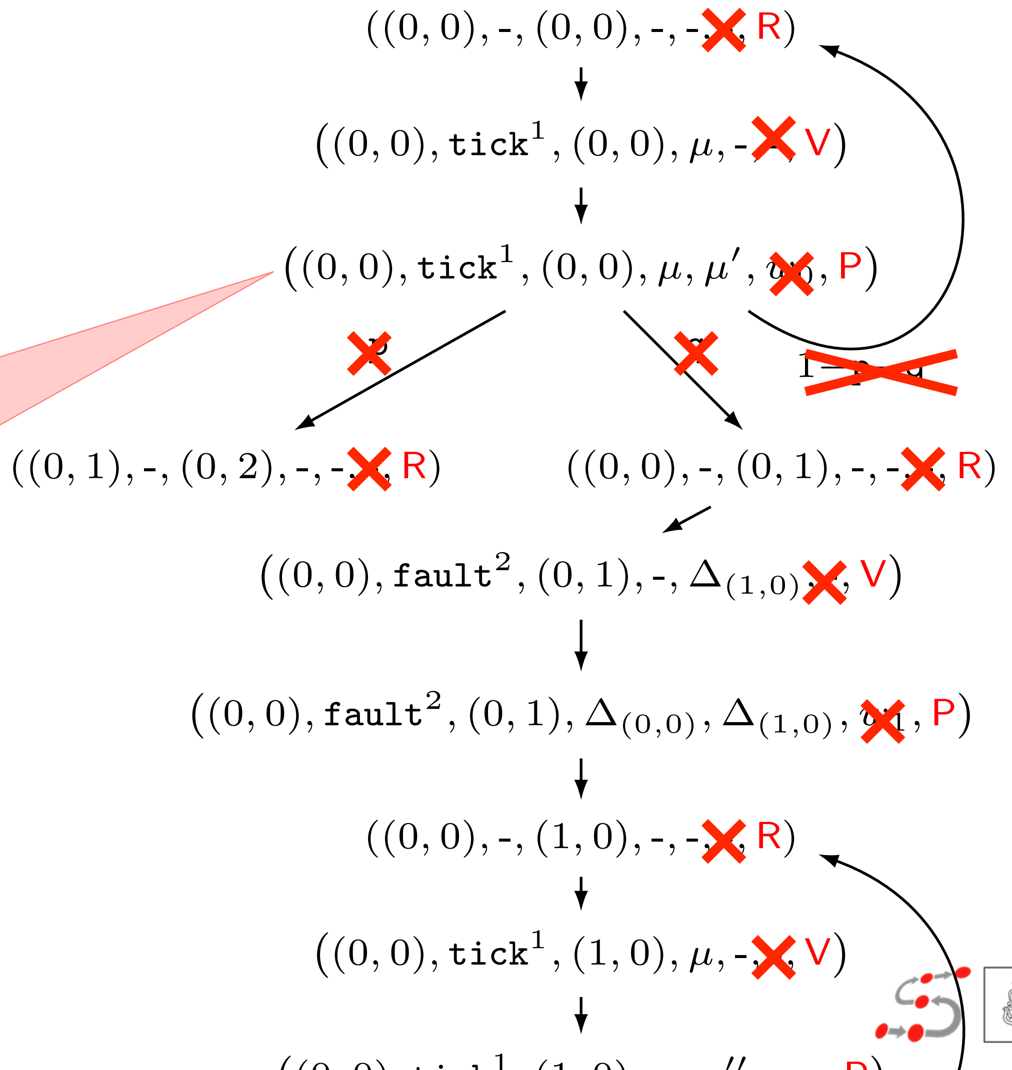
A symbolic (finite!) alternative



A characterizing stochastic game

A symbolic (finite!) alternative

$$x_{(0,1),(0,2)} + x_{(0,1),(0,1)} + x_{(0,1),(0,0)} = \mathtt{p}$$
$$x_{(0,0),(0,2)} + x_{(0,0),(0,1)} + x_{(0,0),(0,0)} = 1 - \mathtt{p}$$
$$x_{(0,1),(0,2)} + x_{(0,0),(0,2)} = \mathtt{p}$$
$$x_{(0,1),(0,0)} + x_{(0,0),(0,0)} = 1 - \mathtt{p} - \mathtt{q}$$
$$x_{(0,1),(0,1)} + x_{(0,0),(0,1)} = \mathtt{q}$$
$$x_{(0,1),(0,2)} \geq 0$$
$$x_{(0,1),(0,1)} \geq 0$$
$$x_{(0,1),(0,0)} \geq 0$$
$$x_{(0,0),(0,2)} \geq 0$$
$$x_{(0,0),(0,1)} \geq 0$$
$$x_{(0,0),(0,0)} \geq 0$$



CONICET

UNC

A characterizing stochastic game

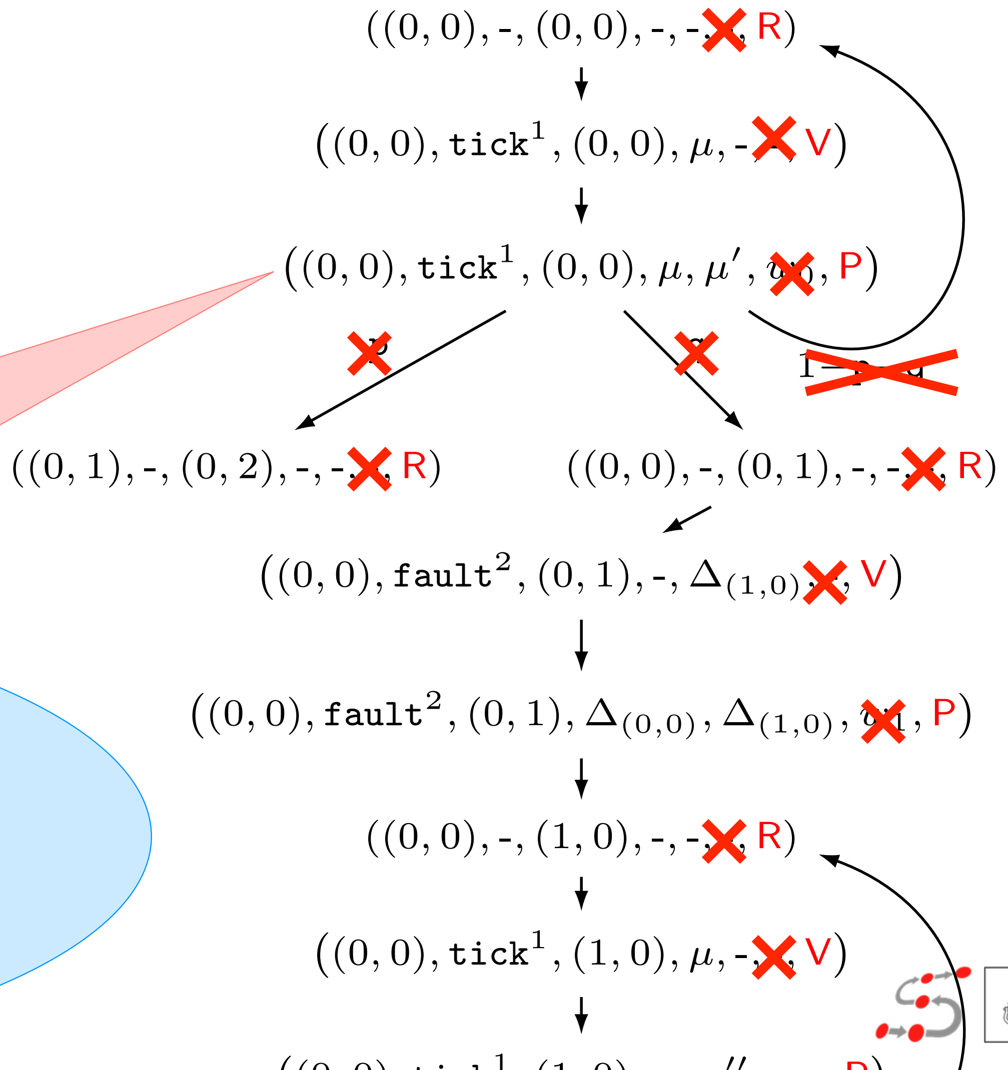
A symbolic (finite!) alternative

$$x_{(0,1),(0,2)} + x_{(0,1),(0,1)} + x_{(0,1),(0,0)} = \mathtt{p}$$
$$x_{(0,0),(0,2)} + x_{(0,0),(0,1)} + x_{(0,0),(0,0)} = 1 - \mathtt{p}$$
$$x_{(0,1),(0,2)} + x_{(0,0),(0,2)} = \mathtt{p}$$
$$x_{(0,1),(0,0)} + x_{(0,0),(0,0)} = 1 - \mathtt{p} - \mathtt{q}$$
$$x_{(0,1),(0,1)} + x_{(0,0),(0,1)} = \mathtt{q}$$
$$x_{(0,1),(0,2)} \geq 0$$

The stochastic game can be solved using the symbolic game using the limit of:

$$U^0 = \{v_{\text{err}}\}$$

$$U^{i+1} = \{v' \mid v' \in V_{\mathsf{R}}^{\mathcal{SG}} \wedge Post^{\mathcal{SG}}(v') \cap (\textstyle\bigcup_{j \leq i} U^j) \neq \emptyset\} \cup \{v' \mid v' \in V_{\mathsf{V}}^{\mathcal{SG}} \wedge Post^{\mathcal{SG}}(v') \subseteq \textstyle\bigcup_{j \leq i} U^j\} \cup \{v' \mid v' \in V_{\mathsf{P}}^{\mathcal{SG}} \wedge Eq(v', Post^{\mathcal{SG}}(v') \cap (\textstyle\bigcup_{j \leq i} U^j)) \text{ has no solution}\}$$



If not masking similar, how close it is?

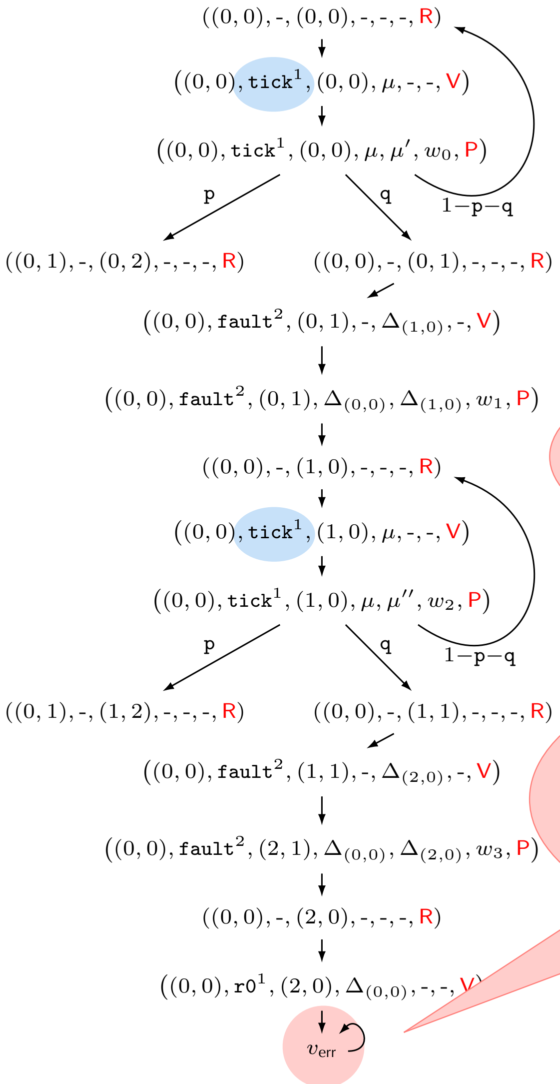
Instead, we consider quantitative objectives:

- Expected total accumulated milestones

  $\mathsf{m} : \Sigma_{\mathcal{F}} \to \mathbb{N}_0$

- The reward is defined by

$$r_{\mathsf{m}}(v) = \begin{cases} \mathsf{m}(\sigma) & \text{if } v \text{ is a V node and } \sigma \text{ the action in it} \\ 0 & \text{otherwise} \end{cases}$$



$\mathsf{m}(\texttt{tick}) = 1$

$\mathsf{m}(a) = 0, \text{ if } a \neq \texttt{tick}$

If not, usually

$$\sup_{\pi_{\mathsf{R}} \in \Pi_{\mathsf{R}}} \inf_{\pi_{\mathsf{V}} \in \Pi_{\mathsf{V}}} Prob^{\pi_{\mathsf{V}}, \pi_{\mathsf{R}}}(\Diamond\, v_{\mathsf{err}}) = 1$$

Probabilistic masking simulation iff

$$\sup_{\pi_{\mathsf{R}} \in \Pi_{\mathsf{R}}} \inf_{\pi_{\mathsf{V}} \in \Pi_{\mathsf{V}}} Prob^{\pi_{\mathsf{V}}, \pi_{\mathsf{R}}}(\Diamond\, v_{\mathsf{err}}) = 0$$

CONICET

UNC

If not masking similar, how close it is?

Instead, we consider quantitative objectives:

- Expected total accumulated milestones

  $\mathsf{m} : \Sigma_{\mathcal{F}} \to \mathbb{N}_0$

- The reward is defined by

$$r_{\mathsf{m}}(v) = \begin{cases} \mathsf{m}(\sigma) & \text{if } v \text{ is a V node and } \sigma \text{ the action in it} \\ 0 & \text{otherwise} \end{cases}$$

- We want to optimize

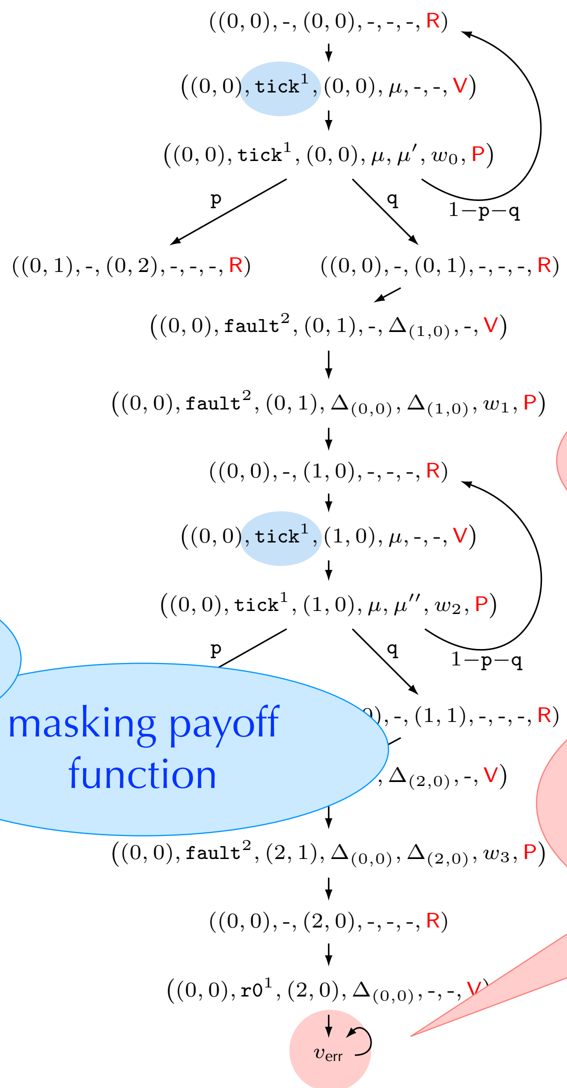
  $\mathbb{E}^{\pi_{\mathsf{V}},\pi_{\mathsf{R}}}[f_{\mathsf{m}}]$

  where

  $f_{\mathsf{m}}(v_0\, v_1\, v_2\, v_3 \ldots) = \lim_{n\to\infty}(\sum_{i=0}^{n} r_{\mathsf{m}}(v_i))$

Expected total reward

masking payoff function



$\mathsf{m}(\mathtt{tick}) = 1$

$\mathsf{m}(a) = 0, \text{ if } a \neq \mathtt{tick}$

If not, usually

$\sup_{\pi_{\mathsf{R}}\in\Pi_{\mathsf{R}}} \inf_{\pi_{\mathsf{V}}\in\Pi_{\mathsf{V}}} Prob^{\pi_{\mathsf{V}},\pi_{\mathsf{R}}}(\Diamond\, v_{\mathsf{err}}) = 1$

Probabilistic masking simulation iff

$\sup_{\pi_{\mathsf{R}}\in\Pi_{\mathsf{R}}} \inf_{\pi_{\mathsf{V}}\in\Pi_{\mathsf{V}}} Prob^{\pi_{\mathsf{V}},\pi_{\mathsf{R}}}(\Diamond\, v_{\mathsf{err}}) = 0$

CONICET

UNC

If not masking similar, how close it is?

In our context:
almost surely failing

Instead, we consider q... objectives:

- Expected total accum...ated milestones

m...

To solve it, the game needs to be stopping: for all pair of strategies the probability of reaching v_{err} is 1

$$r_{\mathsf{m}}(v) = \begin{cases} \ldots & \ldots \text{and } \sigma \text{ the action in it} \\ 0 & \text{otherwise} \end{cases}$$

- We want to optimize

$\mathbb{E}^{\pi_{\mathsf{V}},\pi_{\mathsf{R}}}[f_{\mathsf{m}}]$

where

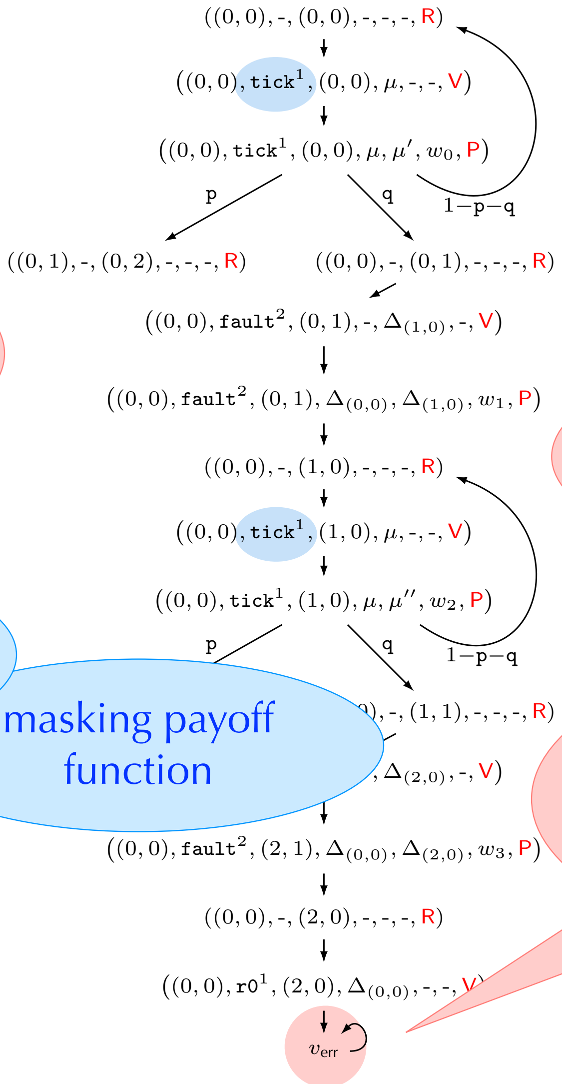
$f_{\mathsf{m}}(v_0\, v_1\, v_2\, v_3 \ldots) = \lim_{n\to\infty}(\sum_{i=0}^{n} r_{\mathsf{m}}(v_i))$

Expected total reward

masking payoff function

$\mathsf{m}(\mathtt{tick}) = 1$

$\mathsf{m}(a) = 0, \text{ if } a \neq \mathtt{tick}$



If not, usually

$$\sup_{\pi_{\mathsf{R}}\in\Pi_{\mathsf{R}}} \inf_{\pi_{\mathsf{V}}\in\Pi_{\mathsf{V}}} Prob^{\pi_{\mathsf{V}},\pi_{\mathsf{R}}}(\Diamond\, v_{\mathsf{err}}) = 1$$

Probabilistic masking simulation iff

$$\sup_{\pi_{\mathsf{R}}\in\Pi_{\mathsf{R}}} \inf_{\pi_{\mathsf{V}}\in\Pi_{\mathsf{V}}} Prob^{\pi_{\mathsf{V}},\pi_{\mathsf{R}}}(\Diamond\, v_{\mathsf{err}}) = 0$$

The need of fairness

```
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
```

CONICET

UNC

The need of fairness

```
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
```

R choses to read

```
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
```

The need of fairness

```
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
```

V imitates with the only choice

```
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
```

CONICET

UNC

The need of fairness

```
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
```

R choses to read

```
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
```

The need of fairness

```
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
```

V imitates with the only choice

```
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
```

CONICET

UNC

The need of fairness

```
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
```

R choses to read

```
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
```

The need of fairness

```
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
```

V imitates with the only choice

We then request that the game is almost sure failing *under fairness*

$$\inf_{\pi_V \in \Pi_V} \inf_{\pi_R \in \Pi_R^f} Prob^{\pi_V, \pi_R}(\Diamond v_{err}) = 1$$

The need of fairness

```
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)          ->
 [w1]   (s!=2)          ->
 [r0]   (s!=2) & (v<=1) ->
 [r1]   (s!=2) & (v>=2) -
 [tick] (s!=2)          -> p: (s'= 2) + q: (s'= 1)
                          (1-p-q): true;

                                   ? 0 : 3);
                                   -1) : 2)
                                           ;
                                 (v-1) : 1)
                                           ;

endmodule
```

…if the game is finite

V imitates with the only choice

In this more general case, stochastic games with expected total reward objectives are determined and can be solved [CAV 2022]

We then request that the game is almost sure failing *under fairness*

$$\inf_{\pi_V \in \Pi_V} \inf_{\pi_R \in \Pi_R^f} Prob^{\pi_V, \pi_R}(\Diamond v_{err}) = 1$$

CONICET

UNC

Generalization to our infinite setting

Theorem:

Stochastic masking games with masking payoff objectives are determined and can be solved

Generalization to our infinite setting

Theorem:

Stochastic masking games with masking payoff objectives are determined and can be solved

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

Generalization to our infinite setting

Theorem:

Stochastic masking games with masking payoff objectives are determined and can be solved

$$\inf_{\pi_R \in \Pi^f_{R,\mathcal{G}}} \sup_{\pi_V \in \Pi_{V,\mathcal{G}}} \mathbb{E}^{\pi_V,\pi_R}_{\mathcal{G},v}[f_m] = \inf_{\pi_R \in \Pi^{MDf}_{R,\mathcal{H}}} \sup_{\pi_V \in \Pi^{MD}_{V,\mathcal{H}}} \mathbb{E}^{\pi_V,\pi_R}_{\mathcal{H},v}[f_m] =$$

$$= \sup_{\pi_V \in \Pi^{MD}_{V,\mathcal{H}}} \inf_{\pi_R \in \Pi^{MDf}_{R,\mathcal{H}}} \mathbb{E}^{\pi_V,\pi_R}_{\mathcal{H},v}[f_m] = \sup_{\pi_V \in \Pi_{V,\mathcal{G}}} \inf_{\pi_R \in \Pi^f_{R,\mathcal{G}}} \mathbb{E}^{\pi_V,\pi_R}_{\mathcal{G},v}[f_m]$$

Generalization to our infinite setti

Provided $\mathcal{H}$ is almost sure failing under fairness

Theorem:

Stochastic masking games with masking payoff objectives are determined and can be solved

$$\inf_{\pi_R \in \Pi^{f}_{R,\mathcal{G}}} \sup_{\pi_V \in \Pi_{V,\mathcal{G}}} \mathbb{E}^{\pi_V,\pi_R}_{\mathcal{G},v}[f_m] = \inf_{\pi_R \in \Pi^{MDf}_{R,\mathcal{H}}} \sup_{\pi_V \in \Pi^{MD}_{V,\mathcal{H}}} \mathbb{E}^{\pi_V,\pi_R}_{\mathcal{H},v}[f_m] =$$

$$= \sup_{\pi_V \in \Pi^{MD}_{V,\mathcal{H}}} \inf_{\pi_R \in \Pi^{MDf}_{R,\mathcal{H}}} \mathbb{E}^{\pi_V,\pi_R}_{\mathcal{H},v}[f_m] = \sup_{\pi_V \in \Pi_{V,\mathcal{G}}} \inf_{\pi_R \in \Pi^{f}_{R,\mathcal{G}}} \mathbb{E}^{\pi_V,\pi_R}_{\mathcal{G},v}[f_m]$$

$\mathcal{H}$ is a finite stochastic game

Applies results in [CAV 2022]

Generalization to our infinite setti…

Provided $\mathcal{H}$ is almost sure failing under fairness

Theorem:

Stochastic masking games with masking payoff objectives are determined and can be solved

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

Generalization to our infinite setti...

Provided $\mathcal{H}$ is almost sure failing under fairness

Theorem:

Stochastic masking games with masking payoff objectives are determined and can be solved

$$\inf_{\pi_R \in \Pi^{f}_{R,\mathcal{G}}} \sup_{\pi_V \in \Pi_{V,\mathcal{G}}} \mathbb{E}^{\pi_V,\pi_R}_{\mathcal{G},v}[f_m]$$

$$\leq \inf_{\pi_R \in \Pi^{MDf}_{R,\mathcal{G}}} \sup_{\pi_V \in \Pi_{V,\mathcal{G}}} \mathbb{E}^{\pi_V,\pi_R}_{\mathcal{G},v}[f_m]$$

$\Pi^{MDf}_{R,\mathcal{G}} \subseteq \Pi^{f}_{R,\mathcal{G}}$

Memoryless deterministic fair strategies

Generalization to our infinite setti...

Provided $\mathcal{H}$ is almost sure failing under fairness

Theorem:

Stochastic masking games with masking payoff objectives are determined and can be solved

$$\inf_{\pi_R \in \Pi^f_{R,\mathcal{G}}} \sup_{\pi_V \in \Pi_{V,\mathcal{G}}} \mathbb{E}^{\pi_V,\pi_R}_{\mathcal{G},v}[f_m]$$

$$\leq \inf_{\pi_R \in \Pi^{MDf}_{R,\mathcal{G}}} \sup_{\pi_V \in \Pi_{V,\mathcal{G}}} \mathbb{E}^{\pi_V,\pi_R}_{\mathcal{G},v}[f_m]$$

$$= \inf_{\pi_R \in \Pi^{MDf}_{R,\mathcal{G}}} \sup_{\pi_V \in \Pi^S_{V,\mathcal{G}}} \mathbb{E}^{\pi_V,\pi_R}_{\mathcal{G},v}[f_m]$$

Lemma: Fix $\pi_R \in \Pi^S_R$. For any $\pi_V \in \Pi_V$, there is a semi-Markov strategy $\pi^*_V \in \Pi^S_V$ s.t.

$$\mathbb{E}^{\pi_V,\pi_R}_{\mathcal{G},v}[f_m] = \mathbb{E}^{\pi^*_V,\pi_R}_{\mathcal{G},v}[f_m]$$

Semi-Markov strategies

if $|\rho| = |\rho'|$ and $\text{last}(\rho) = \text{last}(\rho')$ then $\pi_V(\rho) = \pi_V(\rho')$

CONICET

UNC

Generalization to our infinite setti…

Provided $\mathcal{H}$ is almost sure failing under fairness

Theorem:

Stochastic masking games with masking payoff objectives are determined and can be solved

$$\inf_{\pi_R \in \Pi^{f}_{R,\mathcal{G}}} \sup_{\pi_V \in \Pi_{V,\mathcal{G}}} \mathbb{E}^{\pi_V,\pi_R}_{\mathcal{G},v}[f_m]$$

$$\leq \inf_{\pi_R \in \Pi^{MDf}_{R,\mathcal{G}}} \sup_{\pi_V \in \Pi^{S}_{V,\mathcal{G}}} \mathbb{E}^{\pi_V,\pi_R}_{\mathcal{G},v}[f_m]$$

$$= \inf_{\pi_R \in \Pi^{MDf}_{R,\mathcal{G}}} \sup_{\pi_V \in \Pi^{XS}_{V,\mathcal{G}}} \mathbb{E}^{\pi_V,\pi_R}_{\mathcal{G},v}[f_m]$$

Lemma: Fix $\pi_R \in \Pi^S_R$. For any $\pi_V \in \Pi^S_V$ there is an extreme semi-Markov strategy $\pi^*_V \in \Pi^{XS}_V$ s.t.

$$\mathbb{E}^{\pi_V,\pi_R}_{\mathcal{G},v}[f_m] = \mathbb{E}^{\pi^*_V,\pi_R}_{\mathcal{G},v}[f_m]$$

Extreme semi-Markov strategies

$\pi_V(\rho)((s, \text{-}, s', \mu, \mu', w, P)) > 0$ implies $w \in \mathbb{V}(\mathbb{C}(\mu, \mu'))$

Generalization to our infinite setti

Theorem:

Stochastic masking games with masking payoff objectives are determined and can be solved

$$\inf_{\pi_R \in \Pi^f_{R,\mathcal{G}}} \sup_{\pi_V \in \Pi_{V,\mathcal{G}}} \mathbb{E}^{\pi_V,\pi_R}_{\mathcal{G},v}[f_m]$$

$$\leq \inf_{\pi_R \in \Pi^{MDf}_{R,\mathcal{G}}} \sup_{\pi_V \in \Pi^{XS}_{V,\mathcal{G}}} \mathbb{E}^{\pi_V,\pi_R}_{\mathcal{G},v}[f_m]$$

$$= \inf_{\pi_R \in \Pi^{MDf}_{R,\mathcal{H}}} \sup_{\pi_V \in \Pi^{S}_{V,\mathcal{H}}} \mathbb{E}^{\pi_V,\pi_R}_{\mathcal{H},v}[f_m]$$

CONICET

UNC

Generalization to our infinite setti

Provided $\mathcal{H}$ is almost sure failing under fairness

Theorem:

Stochastic masking games with masking payoff objectives are determined and can be solved

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

By Theorem 5 in [CAV 2022]

Generalization to our infinite setting

Provided $\mathcal{H}$ is almost sure failing under fairness

Theorem:

Stochastic masking games with masking payoff objectives are determined and can be solved

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

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

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

By Theorem 5 in [CAV 2022]

CONICET

UNC

Generalization to our infinite setti

Provided $\mathcal{H}$ is almost sure failing under fairness

Theorem:

Stochastic masking games with masking payoff objectives are determined and can be solved

$$
\begin{aligned}
&\inf_{\pi_R \in \Pi^{f}_{R,\mathcal{G}}} \sup_{\pi_V \in \Pi_{V,\mathcal{G}}} \mathbb{E}^{\pi_V,\pi_R}_{\mathcal{G},v}[f_m] \\
&\leq \inf_{\pi_R \in \Pi^{MDf}_{R,\mathcal{H}}} \sup_{\pi_V \in \Pi^{MD}_{V,\mathcal{H}}} \mathbb{E}^{\pi_V,\pi_R}_{\mathcal{H},v}[f_m] \\
&= \sup_{\pi_V \in \Pi^{MD}_{V,\mathcal{H}}} \inf_{\pi_R \in \Pi^{MDf}_{R,\mathcal{H}}} \mathbb{E}^{\pi_V,\pi_R}_{\mathcal{H},v}[f_m] \\
&= \sup_{\pi_V \in \Pi^{MD}_{V,\mathcal{H}}} \inf_{\pi_R \in \Pi^{f}_{R,\mathcal{H}}} \mathbb{E}^{\pi_V,\pi_R}_{\mathcal{H},v}[f_m]
\end{aligned}
$$

By Lema 6 in [CAV 2022]

CONICET

UNC

Generalization to our infinite setti

Provided $\mathcal{H}$ is almost sure failing under fairness

Theorem:

Stochastic masking games with masking payoff objectives are determined and can be solved

$$\inf_{\pi_R \in \Pi^f_{R,\mathcal{G}}} \sup_{\pi_V \in \Pi_{V,\mathcal{G}}} \mathbb{E}^{\pi_V,\pi_R}_{\mathcal{G},v}[f_m]$$

$$\leq \inf_{\pi_R \in \Pi^{MDf}_{R,\mathcal{H}}} \sup_{\pi_V \in \Pi^{MD}_{V,\mathcal{H}}} \mathbb{E}^{\pi_V,\pi_R}_{\mathcal{H},v}[f_m]$$

$$= \sup_{\pi_V \in \Pi^{MD}_{V,\mathcal{H}}} \inf_{\pi_R \in \Pi^{MDf}_{R,\mathcal{H}}} \mathbb{E}^{\pi_V,\pi_R}_{\mathcal{H},v}[f_m]$$

$$= \sup_{\pi_V \in \Pi^{MD}_{V,\mathcal{H}}} \inf_{\pi_R \in \Pi^{f}_{R,\mathcal{H}}} \mathbb{E}^{\pi_V,\pi_R}_{\mathcal{H},v}[f_m]$$

$$= \sup_{\pi_V \in \Pi^{XMD}_{V,\mathcal{G}}} \inf_{\pi_R \in \Pi^{f}_{R,\mathcal{G}}} \mathbb{E}^{\pi_V,\pi_R}_{\mathcal{G},v}[f_m]$$

Extreme semi-Markov strategies

CONICET

UNC

Generalization to our infinite setti...

Provided $\mathcal{H}$ is almost sure failing under fairness

Theorem:

Stochastic masking games with masking payoff objectives are determined and can be solved

$$\inf_{\pi_R \in \Pi^f_{R,\mathcal{G}}} \sup_{\pi_V \in \Pi_{V,\mathcal{G}}} \mathbb{E}^{\pi_V,\pi_R}_{\mathcal{G},v}[f_m]$$

$$\leq \inf_{\pi_R \in \Pi^{MDf}_{R,\mathcal{H}}} \sup_{\pi_V \in \Pi^{MD}_{V,\mathcal{H}}} \mathbb{E}^{\pi_V,\pi_R}_{\mathcal{H},v}[f_m]$$

$$= \sup_{\pi_V \in \Pi^{MD}_{V,\mathcal{H}}} \inf_{\pi_R \in \Pi^{MDf}_{R,\mathcal{H}}} \mathbb{E}^{\pi_V,\pi_R}_{\mathcal{H},v}[f_m]$$

$$= \sup_{\pi_V \in \Pi^{XMD}_{V,\mathcal{G}}} \inf_{\pi_R \in \Pi^f_{R,\mathcal{G}}} \mathbb{E}^{\pi_V,\pi_R}_{\mathcal{G},v}[f_m]$$

$$\leq \sup_{\pi_V \in \Pi_{V,\mathcal{G}}} \inf_{\pi_R \in \Pi^f_{R,\mathcal{G}}} \mathbb{E}^{\pi_V,\pi_R}_{\mathcal{G},v}[f_m]$$

$\Pi^{XMD}_{V,\mathcal{G}} \subseteq \Pi_{V,\mathcal{G}}$

CONICET

UNC

Generalization to our infinite setti...

Provided $\mathcal{H}$ is almost sure failing under fairness

Theorem:

Stochastic masking games with masking payoff objectives are determined and can be solved

$$\inf_{\pi_R \in \Pi^f_{R,\mathcal{G}}} \sup_{\pi_V \in \Pi_{V,\mathcal{G}}} \mathbb{E}^{\pi_V,\pi_R}_{\mathcal{G},v}[f_m]$$

$$\leq \inf_{\pi_R \in \Pi^{MDf}_{R,\mathcal{H}}} \sup_{\pi_V \in \Pi^{MD}_{V,\mathcal{H}}} \mathbb{E}^{\pi_V,\pi_R}_{\mathcal{H},v}[f_m]$$

$$= \sup_{\pi_V \in \Pi^{MD}_{V,\mathcal{H}}} \inf_{\pi_R \in \Pi^{MDf}_{R,\mathcal{H}}} \mathbb{E}^{\pi_V,\pi_R}_{\mathcal{H},v}[f_m]$$

$$\leq \sup_{\pi_V \in \Pi_{V,\mathcal{G}}} \inf_{\pi_R \in \Pi^f_{R,\mathcal{G}}} \mathbb{E}^{\pi_V,\pi_R}_{\mathcal{G},v}[f_m]$$

$$\leq \inf_{\pi_R \in \Pi^f_{R,\mathcal{G}}} \sup_{\pi_V \in \Pi_{V,\mathcal{G}}} \mathbb{E}^{\pi_V,\pi_R}_{\mathcal{G},v}[f_m]$$

Property inf and sup

Generalization to our infinite setti…

Provided $\mathcal{H}$ is almost sure failing under fairness

Theorem:

Stochastic masking games with masking payoff objectives are determined and can be solved

$$\inf_{\pi_R \in \Pi^f_{R,\mathcal{G}}} \sup_{\pi_V \in \Pi_{V,\mathcal{G}}} \mathbb{E}^{\pi_V,\pi_R}_{\mathcal{G},v}[f_m]$$

$$= \inf_{\pi_R \in \Pi^{MDf}_{R,\mathcal{H}}} \sup_{\pi_V \in \Pi^{MD}_{V,\mathcal{H}}} \mathbb{E}^{\pi_V,\pi_R}_{\mathcal{H},v}[f_m]$$

$$= \sup_{\pi_V \in \Pi^{MD}_{V,\mathcal{H}}} \inf_{\pi_R \in \Pi^{MDf}_{R,\mathcal{H}}} \mathbb{E}^{\pi_V,\pi_R}_{\mathcal{H},v}[f_m]$$

$$= \sup_{\pi_V \in \Pi_{V,\mathcal{G}}} \inf_{\pi_R \in \Pi^f_{R,\mathcal{G}}} \mathbb{E}^{\pi_V,\pi_R}_{\mathcal{G},v}[f_m]$$

Q.E.D.

CONICET

UNC

Generalization to our infinite setti

Provided $\mathcal{H}$ is almost sure failing under fairness

Theorem:

Stochastic masking games with masking payoff objectives are determined and can be solved

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

Q.E.D.

Theorem:

The game can be solved on the symbolic game graph using the following Bellman equations

$$
\begin{aligned}
x_v &= \min\Big(\mathbf{U}, \max_{w \in \mathbb{V}(\mathbb{C}(v[3],v[4]))} \sum_{v' \in Post(v)} w(v'[0], v'[2]) \cdot x_{v'}\Big) && \text{if } v \in V^{\mathcal{SG}}_{\mathsf{P}} \\
x_v &= \min\left(\mathbf{U}, r_{\mathsf{m}}(v) + \max\{x_{v'} \mid v' \in Post(v)\}\right) && \text{if } v \in V^{\mathcal{SG}}_{\mathsf{V}} \\
x_v &= \min\left(\mathbf{U}, \min\{x_{v'} \mid v' \in Post(v)\}\right) && \text{if } v \in V^{\mathcal{SG}}_{\mathsf{R}} \setminus \{v_{\mathsf{err}}\} \\
x_v &= 0 && \text{if } v = v_{\mathsf{err}}
\end{aligned}
$$

Generalization to our infinite setti

Provided $\mathcal{H}$ is almost sure failing under fairness

Theorem:

Stochastic masking games with masking payoff objectives are determined and can be solved

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

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

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

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

Q.E.D.

Theorem:

The game can be solved on the symbolic game graph using the following Bellman equations

$$x_v = \min\Big(\mathbf{U}, \max_{w \in \mathbb{V}(\mathbb{C}(v[3],v[4]))} \sum_{v' \in Post(v)} w(v'[0], v'[2]) \cdot x_{v'}\Big) \qquad \text{if } v \in V^{\mathcal{SG}}_{\mathsf{P}}$$

$$x_v = \min\left(\mathbf{U}, r_{\mathsf{m}}(v) + \max\{x_{v'} \mid v' \in Post(v)\}\right) \qquad \text{if } v \in V^{\mathcal{SG}}_{\mathsf{V}}$$

$$x_v = \min\left(\mathbf{U}, \min\{x_{v'} \mid v' \in Post(v)\}\right) \qquad \text{if } v \in V^{\mathcal{SG}}_{\mathsf{R}} \setminus \{v_{\mathsf{err}}\}$$

$$x_v = 0 \qquad \text{if } v = v_{\mathsf{err}}$$

Uses equations on the symbolic game

The proof uses previous theorem and [CAV 2022]

CONICET

UNC

Generalization to our infinite setti...

Provided $\mathcal{H}$ is almost sure failing under fairness

Theorem:

Stochastic masking games with masking payoff objectives are determined and

Can be checked in polynomial time

$$\inf_{\pi_R \in \Pi^f_{R,\mathcal{G}}} \sup_{\pi_V \in \Pi_{V,\mathcal{G}}} \mathbb{E}^{\pi_V,\pi_R}_{\mathcal{G},v}[f_m]$$

$$= \inf_{\pi_R \in \Pi^{MDf}_{R,\mathcal{H}}} \sup_{\pi_V \in \Pi^{MD}_{V,\mathcal{H}}} \mathbb{E}^{\pi_V,\pi_R}_{\mathcal{H},v}[f_m]$$

$$= \sup_{\pi_V \in \Pi^{MD}_{V,\mathcal{H}}} \inf_{\pi_R \in \Pi^{MDf}_{R,\mathcal{H}}} \mathbb{E}^{\pi_V,\pi_R}_{\mathcal{H},v}[f_m]$$

$$= \sup_{\pi_V \in \Pi_{V,\mathcal{G}}} \inf_{\pi_R \in \Pi^f_{R,\mathcal{G}}} \mathbb{E}^{\pi_V,\pi_R}_{\mathcal{G},v}[f_m]$$

Q.E.D.

Theorem:

The game can be solved on the symbolic game graph using the following Bellman equations

$$x_v = \min\left(\mathbf{U}, \max_{w \in \mathbb{V}(\mathbb{C}(v[3],v[4]))} \sum_{v' \in Post(v)} w(v'[0], v'[2]) \cdot x_{v'}\right) \quad \text{if } v \in V_P^{\mathcal{SG}}$$

$$x_v = \min(\mathbf{U}, r_m(v) + \max\{x_{v'} \mid v' \in Post(v)\}) \quad \text{if } v \in V_V^{\mathcal{SG}}$$

$$x_v = \min(\mathbf{U}, \min\{x_{v'} \mid v' \in Post(v)\}) \quad \text{if } v \in V_R^{\mathcal{SG}} \setminus \{v_{err}\}$$

$$x_v = 0 \quad \text{if } v = v_{err}$$

Uses equations on the symbolic game

The proof uses previous theorem and [CAV 2022]

CONICET

UNC

Summary

- Contributions:
  - A stochastic game characterizing masking probabilistic simulation…
  - …and its symbolic version on which the game can be decided
  - A notion of measure for masking fault tolerance through the stochastic game based on milestones and the resulting masking payoff function
    - This game is infinite but…
    - …determined and can be solved using the symbolic game graph…
    - …under the condition of almost sure failing under fairness…
    - …which is only checkable on the vertex snippet stochastic game graph

Summary

- Contributions:
  - A stochastic game characterizing masking probabilistic simulation…
  - …and its symbolic version on which the game can be decided
  - A notion of measure for masking fault tolerance through the stochastic game based on milestones and the resulting masking payoff function
    - This game is infinite but…
    - …determined and can be solved using the symbolic game graph…
    - …under the condition of almost sure failing under fairness…
    - …which is only checkable on the vertex snippet stochastic game graph

There is a prototype tool soon to be reported

If so, is the stochastic masking game also almost sure failing under fairness?

CONICET

Quantifying Masking Fault-Tolerance via Fair Stochastic Games

Pablo F. Castro, Pedro R. D'Argenio, Ramiro Demasi, Luciano Putruele

UN Córdoba - UN Río Cuarto - CONICET

http://www.cs.famaf.unc.edu.ar/~dargenio/

CONICET

EXPRESS/SOS 2023

UNC