Analysis of Highly Reliable Repairable Fault Trees via Simulation

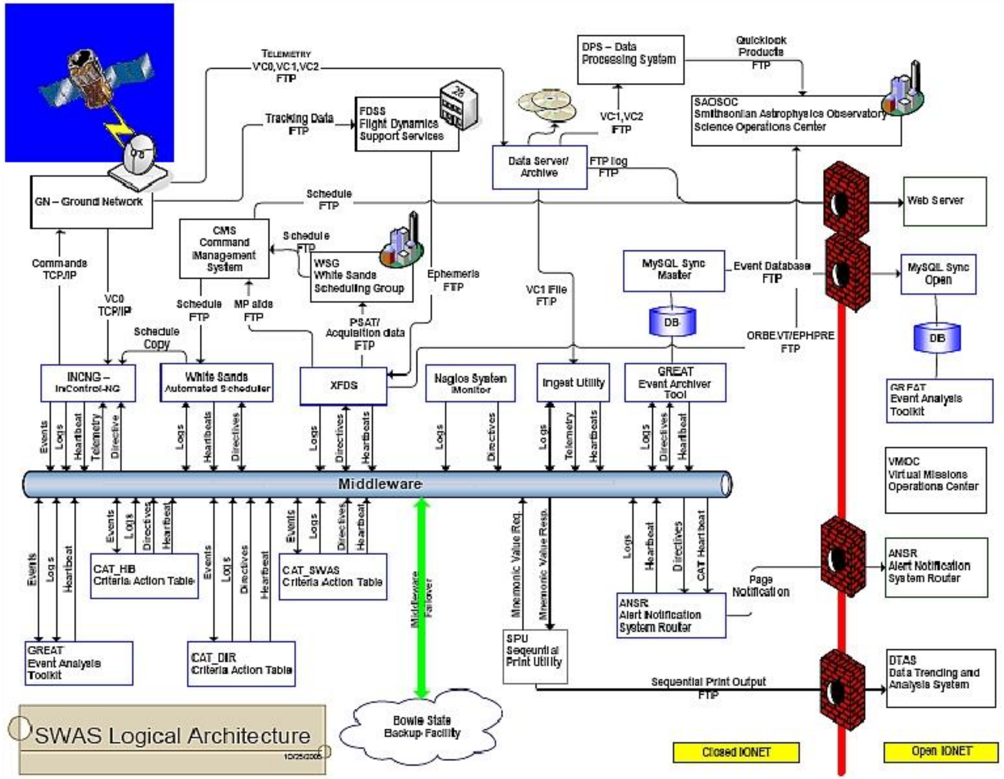
Pedro R. D'Argenio

Universidad Nacional de Córdoba – CONICET (AR)

Joint work with Carlos Budde, Raúl Monti, & Mariëlle Stoelinga

QEST 2022, Warsaw

Fault Tolerant Systems: You know the drill



TELEMETRY VC0,VC1,VC2 FTP
Tracking Data FTP
FDSS Flight Dynamics Support Services
DPS – Data Processing System
Quicklook Products FTP
VC1,VC2 FTP
SAOSOC Smithsonian Astrophysics Observatory Science Operations Center
Data Server/ Archive
FTP log FTP
Web Server
GN – Ground Network
Schedule FTP
CMS Command Management System
Schedule FTP
WSG White Sands Scheduling Group
Ephemeris FTP
VC1 File FTP
MySQL Sync Master
Event Database FTP
MySQL Sync Open
DB
DB
Commands TCP/IP
VC0 TCP/IP
Schedule FTP
MP slide FTP
Schedule Copy
PSAT/ Acquisition data FTP
ORBEVT/EPHPRE FTP
INCNG – InControl-NG
White Sands Automated Scheduler
XFDS
Nagios System Monitor
Ingest Utility
GREAT Event Archiver Tool
GREAT Event Analysis Toolkit
VMOC Virtual Missions Operations Center
Events
Logs
Heartbeat
Telemetry
Directive
Heartbeats
Directives
Middleware
Mnemonic Value Req.
Mnemonic Value Resp.
CAT Heartbeat
CAT_HB Criteria Action Table
CAT_SWAS Criteria Action Table
Middleware Failover
Page Notification
ANSR Alert Notification System Router
ANSR Alert Notification System Router
GREAT Event Analysis Toolkit
CAT_DIR Criteria Action Table
SPU Sequential Print Utility
Sequential Print Output FTP
DTAS Data Trending and Analysis System
SWAS Logical Architecture
10/25/2005
Bowie State Backup Facility
Closed IONET
Open IONET

Failover mechanisms

Voting mechanisms

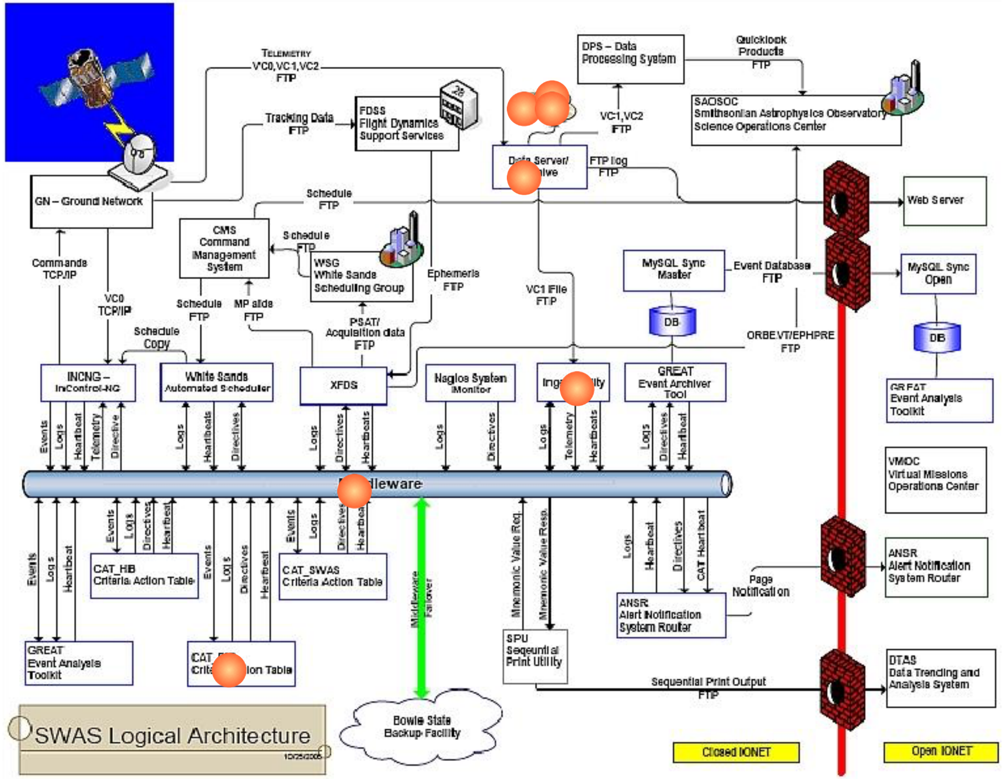
Spare parts

Failsafe mechanisms

Contingency plans

…etc.

Fault Tolerant Systems: You know the drill



Telemetry VC0,VC1,VC2 FTP
Tracking Data FTP
FDSS Flight Dynamics Support Services
DPS – Data Processing System
Quicklook Products FTP
VC1,VC2 FTP
SAOSOC Smithsonian Astrophysics Observatory Science Operations Center
Data Server/ Archive
FTP log FTP
GN – Ground Network
Schedule FTP
Web Server
CMS Command Management System
Schedule FTP
WSG White Sands Scheduling Group
Ephemeris FTP
VC1 File FTP
MySQL Sync Master
Event Database FTP
MySQL Sync Open
Commands TCP/IP
VC0 TCP/IP
Schedule FTP
MP alds FTP
Schedule Copy
PSAT/ Acquisition data FTP
DB
ORBEVT/EPHPRE FTP
DB
INCNG – InControl-NG
White Sands Automated Scheduler
XFDS
Nagios System Monitor
GREAT Event Archiver Tool
GREAT Event Analysis Toolkit
Events
Logs
Heartbeat
Telemetry
Directive
Directives
Heartbeats
Middleware
VMOC Virtual Missions Operations Center
CAT_HB Criteria Action Table
CAT_SWAS Criteria Action Table
Middleware Failover
Mnemonic Value Req.
Mnemonic Value Resp.
CAT Heartbeat
Page Notification
ANSR Alert Notification System Router
ANSR Alert Notification System Router
GREAT Event Analysis Toolkit
CAT Criteria Action Table
SPU Sequential Print Utility
Sequential Print Output FTP
DTAS Data Trending and Analysis System
SWAS Logical Architecture
Bowie State Backup Facility
Closed IONET
Open IONET

Failover mechanisms

Voting mechanisms

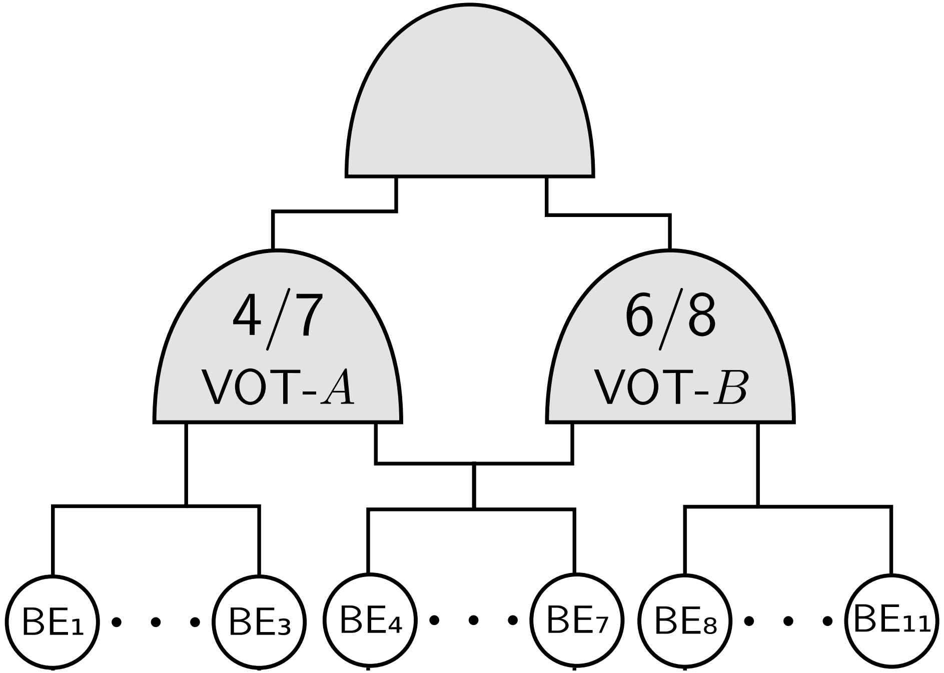
Spare parts

Failsafe mechanisms

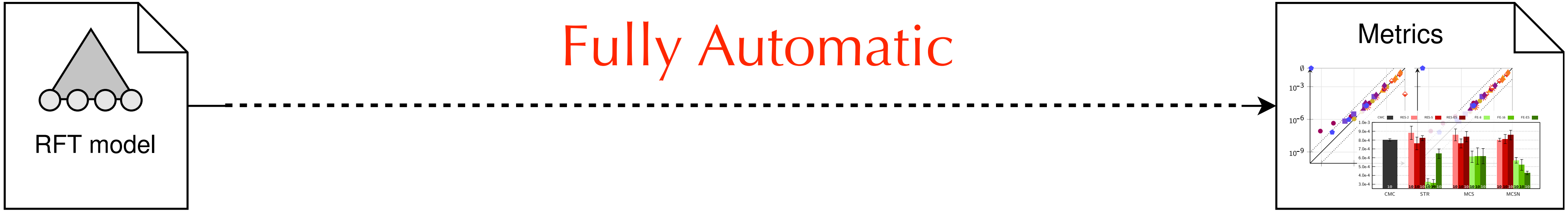
Contingency plans

…etc.

Fault Tolerant Systems: You know the drill



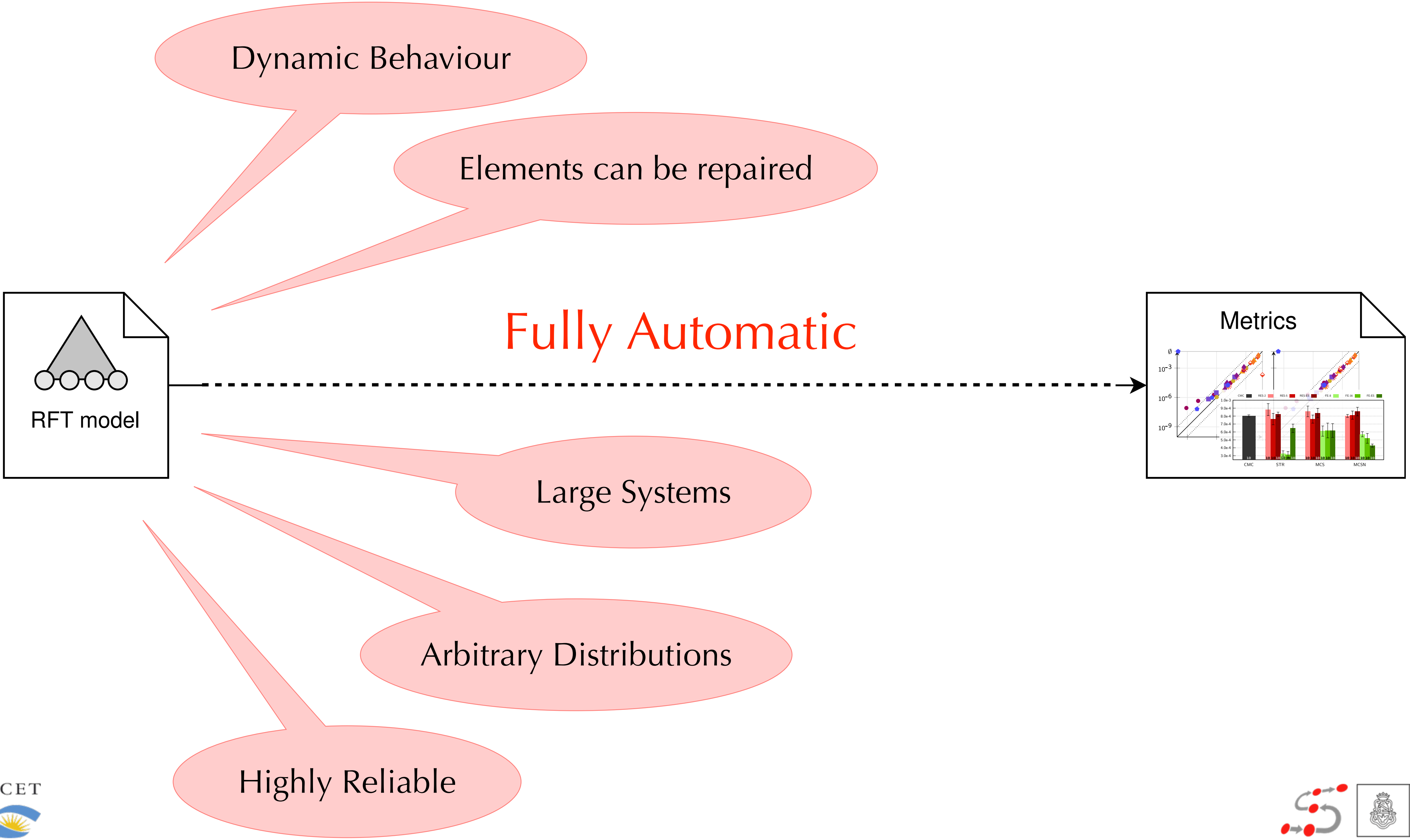
Fault Tree
Analysis



RFT model

Fully Automatic

Metrics



Dynamic Behaviour

Elements can be repaired

Fully Automatic

RFT model

Metrics

Large Systems

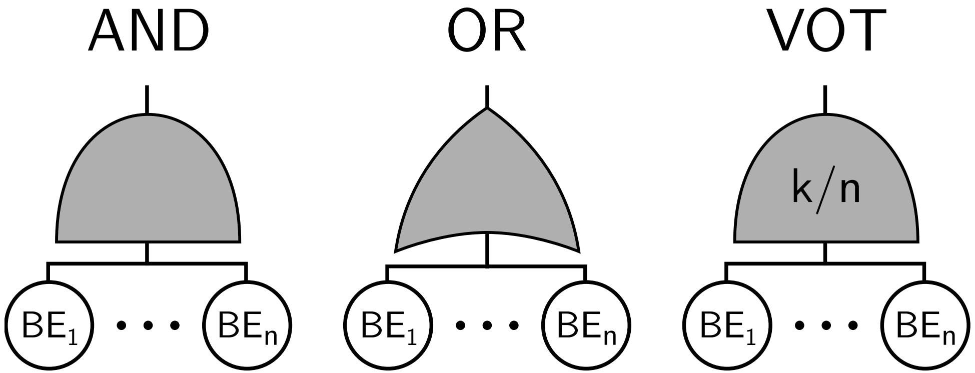
Arbitrary Distributions

Highly Reliable

CONICET

UNC

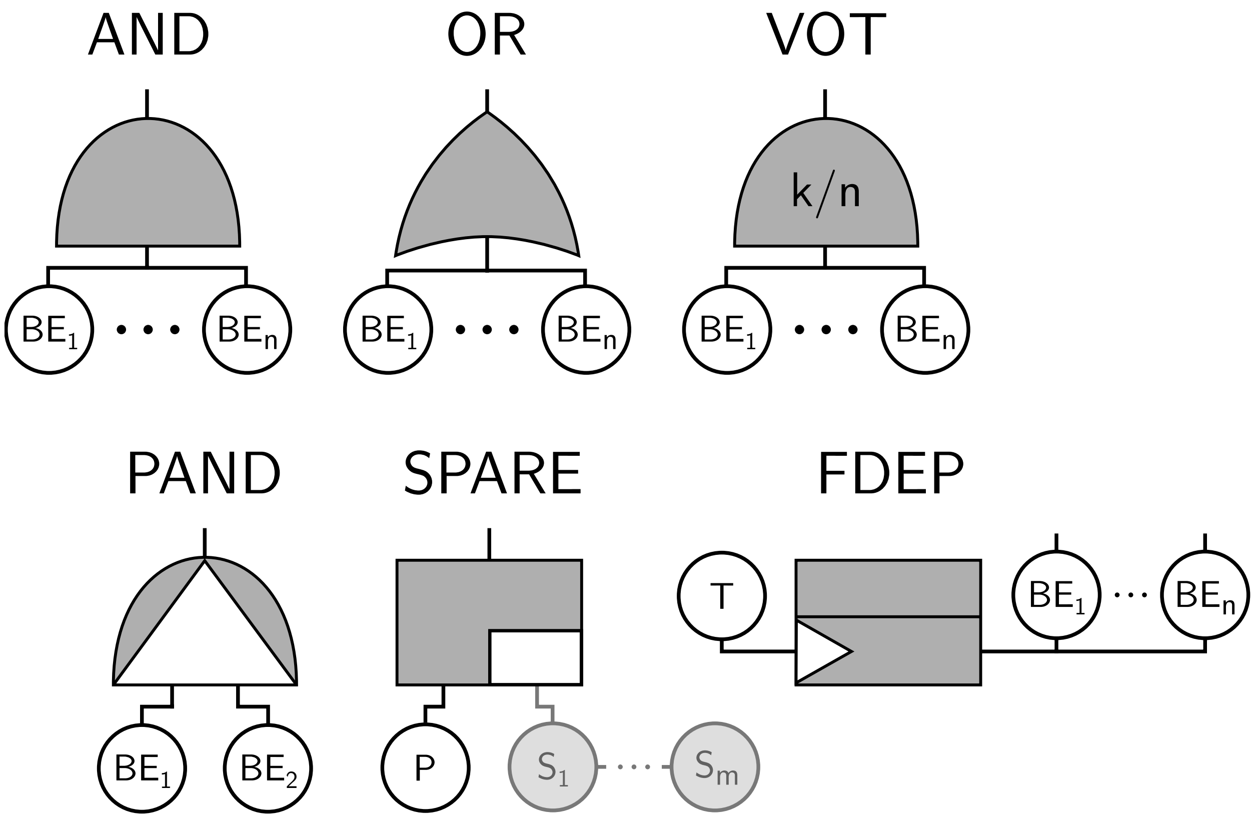
(Static) Fault Trees



Boolean semantics

Dynamic Fault Trees

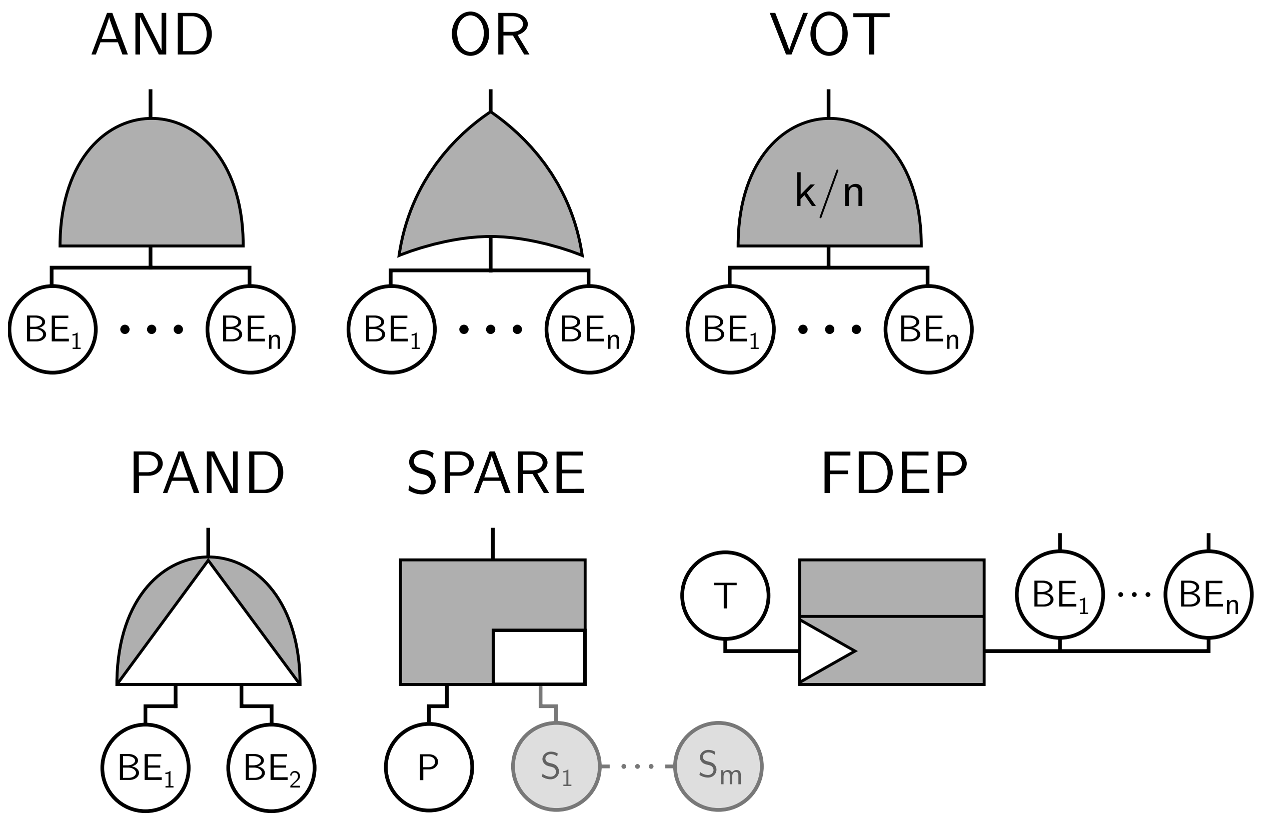
Dynamic Behaviour

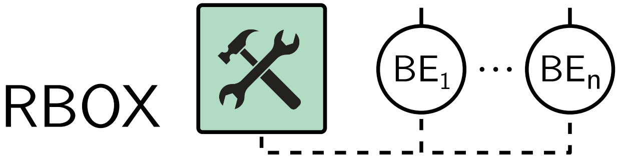


Have a notion of state

Repairable Fault Trees

Elements can be repaired





Have a notion of state

Includes cyclic behaviour

RFT are described in KEPLER

(an extension of GALILEO)

```
toplevel "FAIL";
"FAIL" and "S1" "S2";
"S1" or "SS1" "PS1";
"S2" or "SS2" "PS2";
"SS1" pand "SW1" "M1";
"PS1" sg "M1" "AUX";
"SS2" pand "SW2" "M2";
"PS2" sg "M2" "AUX";
"M1" exponential(0.01) uniform(1,5);
"M2" exponential(0.01) uniform(1,5);
"AUX" exponential(0.01) exponential(0.0025) uniform(1,5);
"SW1" exponential(0.003) uniform(1,2);
"SW2" exponential(0.003) uniform(1,2);
"RBOX" priority_rbox "M1" "M2" "SW1" "SW2" "AUX";
```

Semantics of RFT

Arbitrary Distributions

Large Systems

Semantics of RFT

Arbitrary Distributions

Excludes
Markov Chains

Large Systems

Semantics of RFT

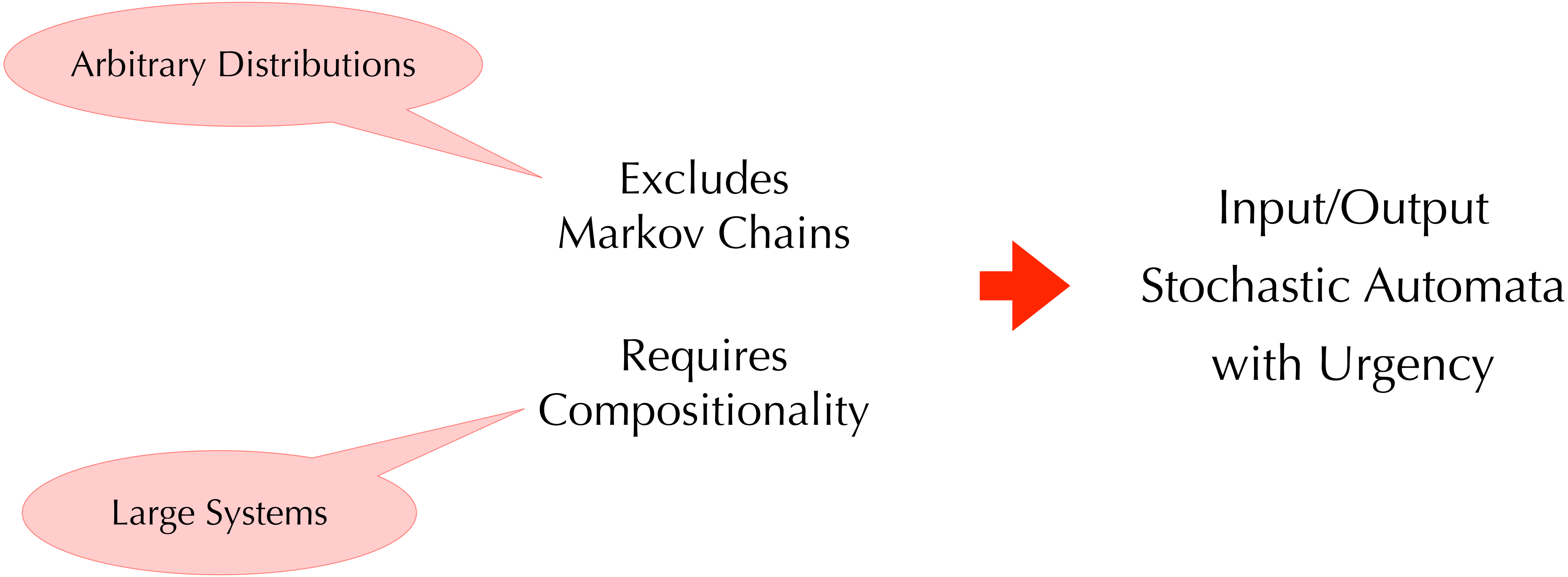
Arbitrary Distributions

Excludes
Markov Chains

Requires
Compositionality

Large Systems

Semantics of RFT



Arbitrary Distributions

Excludes Markov Chains

Requires Compositionality

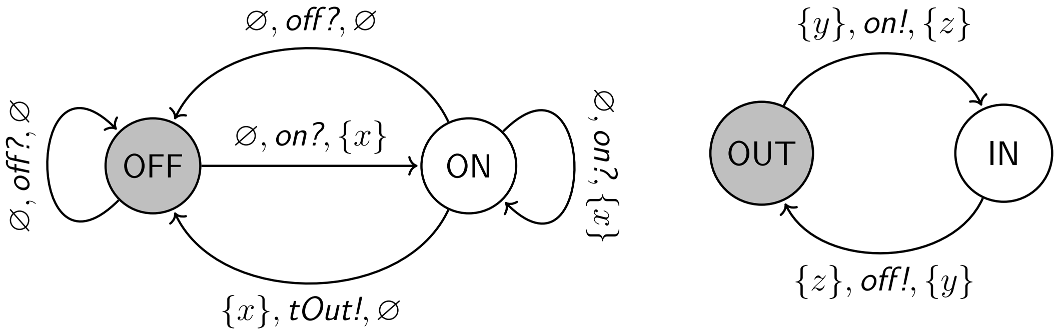
Large Systems

Input/Output Stochastic Automata with Urgency

IOSA + Urgency

$$(\mathcal{S}, \mathcal{A}, \mathcal{C}, \rightarrow, C_0, s_0)$$

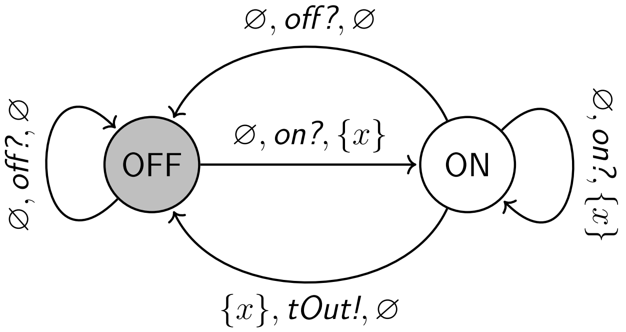
- $\mathcal{S}$ is a set of states
- $\mathcal{A}$ is a set of labels $\begin{cases} \mathcal{A} = \mathcal{A}^{\mathsf{i}} \uplus \mathcal{A}^{\mathsf{o}} \\ \mathcal{A}^{\mathsf{u}} \subseteq \mathcal{A} \end{cases}$
- $\mathcal{C}$ is a set of clocks and each $x \in \mathcal{C}$ has an asociated CDF μ_x
- $\rightarrow \subseteq \mathcal{S} \times \mathcal{C} \times \mathcal{A} \times \mathcal{C} \times S$

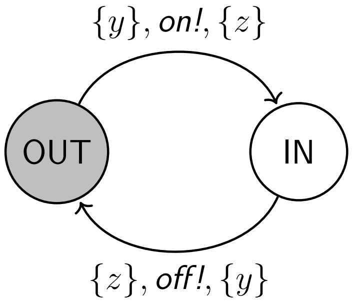

$\varnothing$, *off?*, $\varnothing$
$\varnothing$, *off?*, $\varnothing$
OFF
$\varnothing$, *on?*, $\{x\}$
ON
$\varnothing$, *on?*, $\{x\}$
$\{x\}$, *tOut!*, $\varnothing$
$\{y\}$, *on!*, $\{z\}$
OUT
IN
$\{z\}$, *off!*, $\{y\}$

IOSA + Urgency

$$(\mathcal{S}, \mathcal{A}, \mathcal{C}, \rightarrow, C_0, s_0)$$

- $\mathcal{S}$ is a set of states
- $\mathcal{A}$ is a set of labels $\begin{cases} \mathcal{A} = \mathcal{A}^{\mathsf{i}} \uplus \mathcal{A}^{\mathsf{o}} \\ \mathcal{A}^{\mathsf{u}} \subseteq \mathcal{A} \end{cases}$
- $\mathcal{C}$ is a set of clocks and each $x \in \mathcal{C}$ has an asociated CDF μ_x
- $\rightarrow \subseteq \mathcal{S} \times \mathcal{C} \times \mathcal{A} \times \mathcal{C} \times S$


OFF
ON
$\varnothing$, *off?*, $\varnothing$
$\varnothing$, *off?*, $\varnothing$
$\varnothing$, *on?*, $\{x\}$
$\varnothing$, *on?*, $\{x\}$
$\{x\}$, *tOut!*, $\varnothing$



$$\frac{s_1 \xrightarrow{C,a,C'}_1 s_1'}{s_1||s_2 \xrightarrow{C,a,C'} s_1'||s_2} \; a\in(\mathcal{A}_1 \backslash \mathcal{A}_2)$$

$$\frac{s_1 \xrightarrow{C_1,a,C_1'}_1 s_1' \quad s_2 \xrightarrow{C_2,a,C_2'}_2 s_2'}{s_1||s_2 \xrightarrow{C_1 \cup C_2,a,C_1' \cup C_2'} s_1'||s_2'} \; a\in(\mathcal{A}_1 \cap \mathcal{A}_2)$$

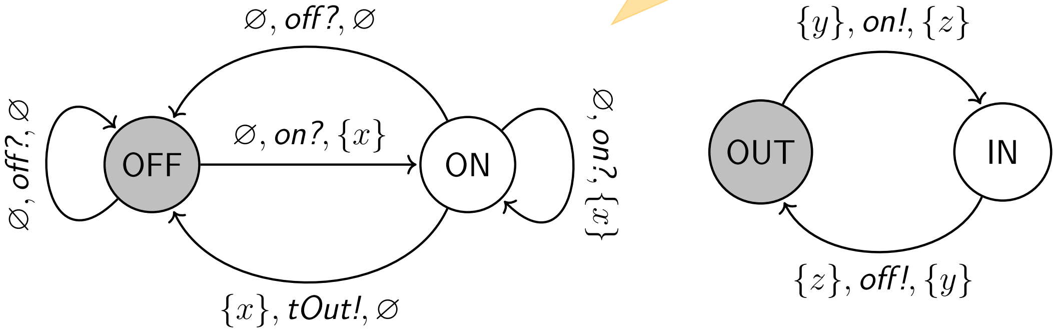
provided $\begin{cases} \mathcal{A}_1^{\mathsf{o}} \cap \mathcal{A}_2^{\mathsf{o}} = \varnothing \\ \mathcal{C}_1 \cap \mathcal{C}_2 = \varnothing \\ \mathcal{A}_1 \cap \mathcal{A}_2^{\mathsf{u}} = \mathcal{A}_2 \cap \mathcal{A}_1^{\mathsf{u}} \end{cases}$

IOSA + Urgency

Analysis through simulation

$$(\mathcal{S}, \mathcal{A}, \mathcal{C}, \rightarrow, C_0, s_0)$$

- $\mathcal{S}$ is a set of states
- $\mathcal{A}$ is a set of labels $\begin{cases} \mathcal{A} = \mathcal{A}^{\mathsf{i}} \uplus \mathcal{A}^{\mathsf{o}} \\ \mathcal{A}^{\mathsf{u}} \subseteq \mathcal{A} \end{cases}$
- $\mathcal{C}$ is a set of clocks and each $x \in \mathcal{C}$ has an asociated CDF μ_x
- $\rightarrow \subseteq \mathcal{S} \times \mathcal{C} \times \mathcal{A} \times \mathcal{C} \times S$


$\varnothing$, off?, $\varnothing$
$\varnothing$, on?, $\{x\}$
$\{x\}$, tOut!, $\varnothing$
$\varnothing$, off?, $\varnothing$
$\varnothing$, on?, $\{x\}$
OFF
ON
$\{y\}$, on!, $\{z\}$
OUT
IN
$\{z\}$, off!, $\{y\}$

$$\frac{s_1 \xrightarrow{C,a,C'}_1 s_1'}{s_1||s_2 \xrightarrow{C,a,C'} s_1'||s_2}\ a\in(\mathcal{A}_1\setminus\mathcal{A}_2)$$

$$\frac{s_1 \xrightarrow{C_1,a,C_1'}_1 s_1' \qquad s_2 \xrightarrow{C_2,a,C_2'}_2 s_2'}{s_1||s_2 \xrightarrow{C_1\cup C_2,a,C_1'\cup C_2'} s_1'||s_2'}\ a\in(\mathcal{A}_1\cap\mathcal{A}_2)$$

provided $\begin{cases} \mathcal{A}_1^{\mathsf{o}} \cap \mathcal{A}_2^{\mathsf{o}} = \varnothing \\ \mathcal{C}_1 \cap \mathcal{C}_2 = \varnothing \\ \mathcal{A}_1 \cap \mathcal{A}_2^{\mathsf{u}} = \mathcal{A}_2 \cap \mathcal{A}_1^{\mathsf{u}} \end{cases}$

IOSA: weak determinism

An IOSA should satisfy:

(a) If $s \xrightarrow{C,a,C'} s'$ and $a \in \mathcal{A}^{\mathsf{i}} \cup \mathcal{A}^{\mathsf{u}}$, then $C = \varnothing$.

(b) If $s \xrightarrow{C,a,C'} s'$ and $a \in \mathcal{A}^{\mathsf{o}} \setminus \mathcal{A}^{\mathsf{u}}$, then C is a singleton set.

(c) If $s \xrightarrow{\{x\},a_1,C_1} s_1$ and $s \xrightarrow{\{x\},a_2,C_2} s_2$ then $a_1 = a_2$, $C_1 = C_2$ and $s_1 = s_2$.

(d) For every $a \in \mathcal{A}^{\mathsf{i}}$ and state s, there exists a transition $s \xrightarrow{\varnothing,a,C} s'$.

(e) For every $a \in \mathcal{A}^{\mathsf{i}}$, if $s \xrightarrow{\varnothing,a,C'_1} s_1$ and $s \xrightarrow{\varnothing,a,C'_2} s_2$, $C'_1 = C'_2$ and $s_1 = s_2$.

(f) There exists a function $\mathrm{active} : \mathcal{S} \to 2^{\mathcal{C}}$ such that:

- (i) $\mathrm{active}(s_0) \subseteq C_0$,
- (ii) $\mathrm{enabling}(s) \subseteq \mathrm{active}(s)$,
- (iii) if s is stable, $\mathrm{active}(s) = \mathrm{enabling}(s)$, and
- (iv) if $t \xrightarrow{C,a,C'} s$ then $\mathrm{active}(s) \subseteq (\mathrm{active}(t) \setminus C) \cup C'$.

IOSA: weak determinism

An IOSA should satisfy:

(a) If $s \xrightarrow{C,a,C'} s'$ and $a \in \mathcal{A}^{\mathsf{i}} \cup \mathcal{A}^{\mathsf{u}}$, then $C = \varnothing$.

(b) If $s \xrightarrow{C,a,C'} s'$ and $a \in \mathcal{A}^{\mathsf{o}} \setminus \mathcal{A}^{\mathsf{u}}$, then C is a singleton set.

(c) If $s \xrightarrow{\{x\},a_1,C_1} s_1$ and $s \xrightarrow{\{x\},a_2,C_2} s_2$ then $a_1 = a_2$, $C_1 = C_2$ and $s_1 = s_2$.

(d) For every $a \in \mathcal{A}^{\mathsf{i}}$ and state s, there exists a transition $s \xrightarrow{\varnothing,a,C} s'$.

Input enabledness

(e) For every $a \in \mathcal{A}^{\mathsf{i}}$, if $s \xrightarrow{\varnothing,a,C_1'} s_1$ and $s \xrightarrow{\varnothing,a,C_2'} s_2$, $C_1' = C_2'$ and $s_1 = s_2$.

(f) There exists a function $\mathrm{active} : \mathcal{S} \to 2^{\mathcal{C}}$ such that:

- (i) $\mathrm{active}(s_0) \subseteq C_0$,
- (ii) $\mathrm{enabling}(s) \subseteq \mathrm{active}(s)$,
- (iii) if s is stable, $\mathrm{active}(s) = \mathrm{enabling}(s)$, and
- (iv) if $t \xrightarrow{C,a,C'} s$ then $\mathrm{active}(s) \subseteq (\mathrm{active}(t) \setminus C) \cup C'$.

IOSA: weak determinism

An IOSA should satisfy:

(a) If $s \xrightarrow{C,a,C'} s'$ and $a \in \mathcal{A}^{\mathsf{i}} \cup \mathcal{A}^{\mathsf{u}}$, then $C = \varnothing$.

(b) If $s \xrightarrow{C,a,C'} s'$ and $a \in \mathcal{A}^{\mathsf{o}} \setminus \mathcal{A}^{\mathsf{u}}$, then C is a singleton set.

(c) If $s \xrightarrow{\{x\},a_1,C_1} s_1$ and $s \xrightarrow{\{x\},a_2,C_2} s_2$ then $a_1 = a_2$, $C_1 = C_2$ and $s_1 = s_2$.

(d) For every $a \in \mathcal{A}^{\mathsf{i}}$ and state s, there exists a transition $s \xrightarrow{\varnothing,a,C} s'$.

(e) For every $a \in \mathcal{A}^{\mathsf{i}}$, if $s \xrightarrow{\varnothing,a,C_1'} s_1$ and $s \xrightarrow{\varnothing,a,C_2'} s_2$, $C_1' = C_2'$ and $s_1 = s_2$.

(f) There exists a function $\mathrm{active} : \mathcal{S} \to 2^{\mathcal{C}}$ such that:

(i) $\mathrm{active}(s_0) \subseteq C_0$,

(ii) $\mathrm{enabling}(s) \subseteq \mathrm{active}(s)$,

(iii) if s is stable, $\mathrm{active}(s) = \mathrm{enabling}(s)$, and

(iv) if $t \xrightarrow{C,a,C'} s$ then $\mathrm{active}(s) \subseteq (\mathrm{active}(t) \setminus C) \cup C'$.

Input enabledness

Input and urgent determinism

IOSA: weak determinism

An IOSA should satisfy:

(a) If $s \xrightarrow{C,a,C'} s'$ and $a \in \mathcal{A}^{\mathsf{i}} \cup \mathcal{A}^{\mathsf{u}}$, then $C = \varnothing$.

(b) If $s \xrightarrow{C,a,C'} s'$ and $a \in \mathcal{A}^{\mathsf{o}} \setminus \mathcal{A}^{\mathsf{u}}$, then C is a singleton set.

(c) If $s \xrightarrow{\{x\},a_1,C_1} s_1$ and $s \xrightarrow{\{x\},a_2,C_2} s_2$ then $a_1 = a_2$, $C_1 = C_2$ and $s_1 = s_2$.

(d) For every $a \in \mathcal{A}^{\mathsf{i}}$ and state s, there exists a transition $s \xrightarrow{\varnothing,a,C} s'$.

(e) For every $a \in \mathcal{A}^{\mathsf{i}}$, if $s \xrightarrow{\varnothing,a,C_1'} s_1$ and $s \xrightarrow{\varnothing,a,C_2'} s_2$, $C_1' = C_2'$ and $s_1 = s_2$.

(f) There exists a function $\mathrm{active} : \mathcal{S} \to 2^{\mathcal{C}}$ such that:

(i) $\mathrm{active}(s_0) \subseteq C_0$,

(ii) $\mathrm{enabling}(s) \subseteq \mathrm{active}(s)$,

(iii) if s is stable, $\mathrm{active}(s) = \mathrm{enabling}(s)$, and

(iv) if $t \xrightarrow{C,a,C'} s$ then $\mathrm{active}(s) \subseteq (\mathrm{active}(t) \setminus C) \cup C'$.

Output determinism (non-urgent)

Input enabledness

Input and urgent determinism

IOSA: weak determinism

An IOSA should satisfy:

(a) If $s \xrightarrow{C,a,C'} s'$ and $a \in \mathcal{A}^{\mathsf{i}} \cup \mathcal{A}^{\mathsf{u}}$, then $C = \varnothing$.

(b) If $s \xrightarrow{C,a,C'} s'$ and $a \in \mathcal{A}^{\mathsf{o}} \setminus \mathcal{A}^{\mathsf{u}}$, then C is a singleton set.

(c) If $s \xrightarrow{\{x\},a_1,C_1} s_1$ and $s \xrightarrow{\{x\},a_2,C_2} s_2$ then $a_1 = a_2$, $C_1 = C_2$ and $s_1 = s_2$.

(d) For every $a \in \mathcal{A}^{\mathsf{i}}$ and state s, there exists a transition $s \xrightarrow{\varnothing,a,C} s'$.

(e) For every $a \in \mathcal{A}^{\mathsf{i}}$, if $s \xrightarrow{\varnothing,a,C_1'} s_1$ and $s \xrightarrow{\varnothing,a,C_2'} s_2$, $C_1' = C_2'$ and $s_1 = s_2$.

(f) There exists a function $\mathrm{active} : \mathcal{S} \to 2^{\mathcal{C}}$ such that:

- (i) $\mathrm{active}(s_0) \subseteq C_0$,
- (ii) $\mathrm{enabling}(s) \subseteq \mathrm{active}(s)$,
- (iii) if s is stable, $\mathrm{active}(s) = \mathrm{enabling}(s)$, and
- (iv) if $t \xrightarrow{C,a,C'} s$ then $\mathrm{active}(s) \subseteq (\mathrm{active}(t) \setminus C) \cup C'$.

Output determinism (non-urgent)

Input enabledness

Input and urgent determinism

The rest ensures that clocks do not introduce non determinism

IOSA: weak determinism

An IOSA should satisfy:

(a) If $s \xrightarrow{C,a,C'} s'$ and $a \in \mathcal{A}^{\mathsf{i}} \cup \mathcal{A}^{\mathsf{u}}$, then $C = \varnothing$.

(b) If $s \xrightarrow{C,a,C'} s'$ and $a \in \mathcal{A}^{\mathsf{o}} \setminus \mathcal{A}^{\mathsf{u}}$, then C is a singleton set.

(c) If $s \xrightarrow{\{x\},a_1,C_1}$ then $a_1 = a_2$, $C_1 = C_2$ and $s_1 = s_2$.

(d) For ev a transition $s \xrightarrow{\varnothing,a,C} s'$.

(e) For ever $\xrightarrow{,a,C_2'}$ s_2, $C_1' = C_2'$ and $s_1 = s_2$.

(f) There exists a fu $\to 2^{\mathcal{C}}$ such that:

(i) $\text{active}(s_0) \subseteq C_0$,

(ii) $\text{enabling}(s) \subseteq \text{active}(s)$,

(iii) if s is stable, $\text{active}(s) = \text{enabling}(s)$, and

(iv) if $t \xrightarrow{C,a,C'} s$ then $\text{active}(s) \subseteq (\text{active}(t) \setminus C) \cup C'$.

Ensures that non-urgent behaviour is deterministic

Output determinism (non-urgent)

Input enabledness

Input and urgent determinism

The rest ensures that clocks do not introduce non determinism

IOSA: weak determinism

Definition 8. *A closed IOSA $\mathcal{I}$ is* weakly deterministic *if* $\Rightarrow$ *is well defined in* $\mathcal{I}$ *and, in* $\mathbf{P}(\mathcal{I})$*, any state* $(s, v) \in \mathbf{S}$ *that satisfies one of the following conditions is almost never reached from any* $(\mathsf{init}, v_0) \in \mathbf{S}$*: (a)* s *is stable and* $\bigcup_{a \in \mathcal{A} \cup \{\mathsf{init}\}} \mathcal{T}_a(s, v)$ *contains at least two different probability measures, (b)* s *is not stable,* $(s, v) \Rightarrow \mu$*,* $(s, v) \Rightarrow \mu'$ *and* $\mu \neq \mu'$*, or (c)* s *is not stable and* $(s, v) \xrightarrow{a} \mu$ *for some* $a \in \mathcal{A}^{\mathsf{o}} \setminus \mathcal{A}^{\mathsf{u}}$*.*

IOSA: weak determinism

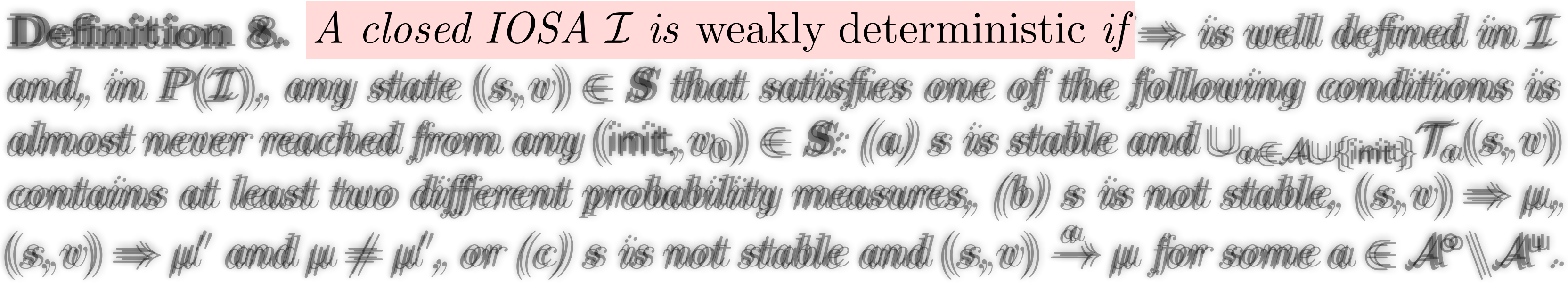
Definition 8. *A closed IOSA $\mathcal{I}$ is* weakly deterministic *if* $\Rightarrow$ *is well defined in* $\mathcal{I}$ *and, in* $P(\mathcal{I})$*, any state* $(s, v) \in \mathbf{S}$ *that satisfies one of the following conditions is almost never reached from any* $(\mathsf{init}, v_0) \in \mathbf{S}$*: (a)* s *is stable and* $\bigcup_{a \in \mathcal{A} \cup \{\mathsf{init}\}} \mathcal{T}_a(s, v)$ *contains at least two different probability measures, (b)* s *is not stable,* $(s, v) \Rightarrow \mu$*,* $(s, v) \Rightarrow \mu'$ *and* $\mu \neq \mu'$*, or (c)* s *is not stable and* $(s, v) \xrightarrow{a} \mu$ *for some* $a \in \mathcal{A}^{\mathsf{o}} \setminus \mathcal{A}^{\mathsf{u}}$.

Theorem: Every closed confluent IOSA is weakly deterministic.

CONICET

UNC

IOSA: weak determinism



Definition 8. *A closed IOSA $\mathcal{I}$ is* weakly deterministic *if* $\Rightarrow$ *is well defined in* $\mathcal{I}$ *and, in* $\mathcal{P}(\mathcal{I})$, *any state* $(s,v)\in\mathbf{S}$ *that satisfies one of the following conditions is almost never reached from any* $(\mathsf{init},v_0)\in\mathbf{S}$*: (a) s is stable and* $\bigcup_{a\in\mathcal{A}\cup\{\mathsf{init}\}}\mathcal{T}_a(s,v)$ *contains at least two different probability measures, (b) s is not stable,* $(s,v)\Rightarrow\mu$, $(s,v)\Rightarrow\mu'$ *and* $\mu\neq\mu'$, *or (c) s is not stable and* $(s,v)\xrightarrow{a}\mu$ *for some* $a\in\mathcal{A}^o\setminus\mathcal{A}^u$.

Theorem: Every closed confluent IOSA is weakly deterministic.

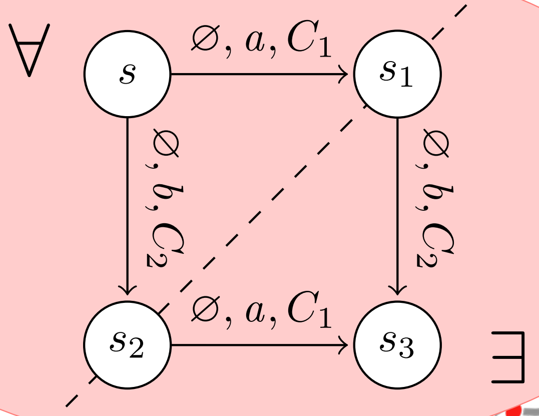
All communications have been resolved (i.e. no inputs left)

IOSA: weak determinism

Definition 8. *A closed IOSA $\mathcal{I}$ is* weakly deterministic *if* $\Rightarrow$ *is well defined in* $\mathcal{I}$ *and, in* $\mathcal{P}(\mathcal{I})$, *any state* $(s,v) \in \mathbf{S}$ *that satisfies one of the following conditions is almost never reached from any* $(\mathsf{init}, v_0) \in \mathbf{S}$: *(a) s is stable and* $\bigcup_{a\in\mathcal{A}\cup\{\mathsf{init}\}} \mathcal{T}_a(s,v)$ *contains at least two different probability measures, (b) s is not stable,* $(s,v) \Rightarrow \mu$, $(s,v) \Rightarrow \mu'$ *and* $\mu \neq \mu'$, *or (c) s is not stable and* $(s,v) \xrightarrow{a} \mu$ *for some* $a \in \mathcal{A}^o \setminus \mathcal{A}^u$.

Theorem: Every closed confluent IOSA is weakly deterministic.

All communications have been resolved (i.e. no inputs left)



IOSA: weak determinism

Definition 8. *A closed IOSA $\mathcal{I}$ is* weakly deterministic *if* $\Rightarrow$ *is well defined in* $\mathcal{I}$ *and, in* $P(\mathcal{I})$*, any state* $(s,v) \in \mathbf{S}$ *that satisfies one of the following conditions is almost never reached from any* $(\mathsf{init}, v_0) \in \mathbf{S}$*: (a)* s *is stable and* $\bigcup_{a \in \mathcal{A} \cup \{\mathsf{init}\}} \mathcal{T}_a(s,v)$ *contains at least two different probability measures, (b)* s *is not stable,* $(s,v) \Rightarrow \mu$*,* $(s,v) \Rightarrow \mu'$ *and* $\mu \neq \mu'$*, or (c)* s *is not stable and* $(s,v) \xrightarrow{a} \mu$ *for some* $a \in \mathcal{A}^{\mathrm{o}} \setminus \mathcal{A}^{\mathrm{u}}$*.*

Theorem: Every closed confluent IOSA is weakly deterministic.

Theorem 5. *Let* $\mathcal{I} = (\mathcal{I}_1 || \cdots || \mathcal{I}_n)$ *be a closed IOSA. If* $\mathcal{I}$ *potentially reaches a non-confluent state then there are actions* $a, b \in \mathcal{A}^{\mathrm{u}} \cap \mathcal{A}^{\mathrm{o}}$ *such that some* $\mathcal{I}_i$ *is not confluent w*[illegible] $c \rightsquigarrow^* a$*,* $d \rightsquigarrow^* b$*, and, either (i)* [illegible] *or (ii) there is some* $e \in \mathcal{A}$ *and (possibly empty) sets* $B_1, \ldots, B_m$ *spontaneously enabled by* e *in* $\mathcal{I}_1, \ldots, \mathcal{I}_m$ *respectively, such that* $c, d \in \bigcup_{i=1}^{m} B_i$*.*

Sufficient conditions for confluency

CONICET

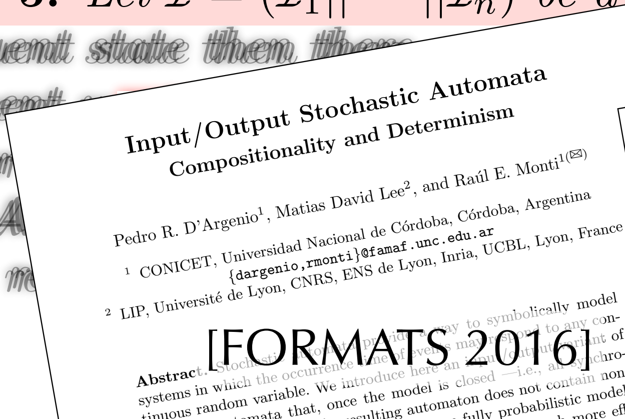
UNC

IOSA: weak determinism

Definition 8. *A closed IOSA $\mathcal{I}$ is* weakly deterministic *if* $\Rightarrow$ *is well defined in* $\mathcal{I}$ *and, in* $P(\mathcal{I})$*, any state* $(s,v) \in \mathbf{S}$ *that satisfies one of the following conditions is almost never reached from any* $(\mathsf{init}, v_0) \in \mathbf{S}$*: (a)* s *is stable and* $\bigcup_{a\in\mathcal{A}\cup\{\mathsf{init}\}} \mathcal{T}_a(s,v)$ *contains at least two different probability measures, (b)* s *is not stable,* $(s,v) \Rightarrow \mu$*,* $(s,v) \Rightarrow \mu'$ *and* $\mu \neq \mu'$*, or (c)* s *is not stable and* $(s,v) \xrightarrow{a} \mu$ *for some* $a \in \mathcal{A}^o \setminus \mathcal{A}^u$.

Theorem: Every closed confluent IOSA is weakly deterministic.

Theorem 5. *Let* $\mathcal{I} = (\mathcal{I}_1||\cdots||\mathcal{I}_n)$ *be a closed IOSA. If* $\mathcal{I}$ *potentially reaches a*
non-confluent state then there … *s* $a,b \in \mathcal{A}^u \cap \mathcal{A}^o$ … *me* $\mathcal{I}_i$ *is*
not confluent … *d* $\rightsquigarrow^* b$*,*
and, either … *there is*
some $e \in \mathcal{A}$ … *by* e *in*
$\mathcal{I}_1,\ldots,\mathcal{I}_n$ *re* …



Input/Output Stochastic Automata
Compositionality and Determinism

Pedro R. D'Argenio[1], Matias David Lee[2], and Raúl E. Monti[1](✉)

[1] CONICET, Universidad Nacional de Córdoba, Córdoba, Argentina
{dargenio,rmonti}@famaf.unc.edu.ar
[2] LIP, Université de Lyon, CNRS, ENS de Lyon, Inria, UCBL, Lyon, France

Abstract. Stochastic automata provide a way to symbolically model systems in which the occurrence time of events may respond to any continuous random variable. We introduce here an input/output variant of automata that, once the model is closed —i.e., all synchro- … resulting automaton does not contain non- … fully probabilistic models … more effi-

[FORMATS 2016]

Check for updates

Input/Output Stochastic Automata with Urgency: Confluence and Weak Determinism

Pedro R. D'Argenio[1,2,3](✉) and Raúl E. Monti[1,2]

[1] Universidad Nacional de Córdoba, FAMAF, Córdoba, Argentina
{dargenio,rmonti}@famaf.unc.edu.ar
[2] CONICET, Córdoba, Argentina
[3] Saarland University, Department of Computer Science, Saarbrücken, Germany

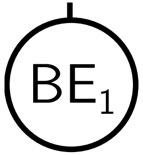
… ious work, we introduced an input/output variant of … ce the model is closed (i.e., all syn- … n is fully stochastic, … vari-

[ICTAC 2018]

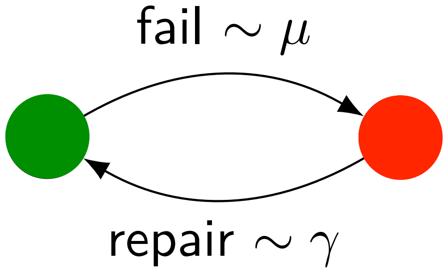
CONICET

UNC

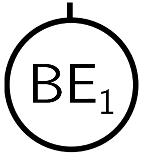
From RFT to IOSA



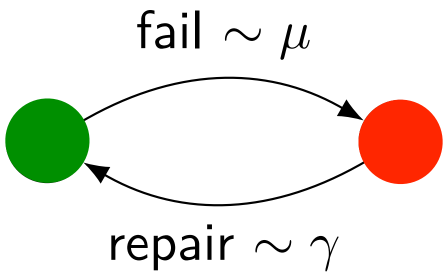
Basic Element



From RFT to IOSA



Basic Element

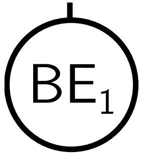


```
module BE_i
  fc, rc : clock;
  inform : [0..2] init 0;
  broken : [0..2] init 0;   // 0: up, 1: down, 2: repairing

  [fl!] broken=0 @ fc -> (inform=1) & (broken=1);
  [r??] broken=1      -> (broken=2) & (rc=γ);
  [up!] broken=2 @ rc -> (inform=2) &
                         (broken=0) & (fc=μ);

  [fi!!] inform=1 -> (inform=0);
  [ui!!] inform=2 -> (inform=0);
endmodule
```

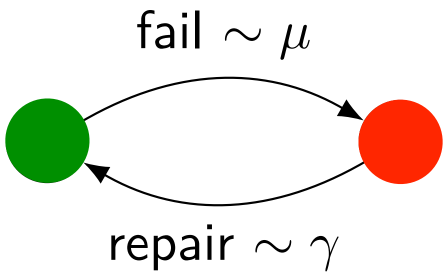
Textual form of IOSA
for the tool FIG

From RFT to IOSA

Assume self-loops for undefined inputs



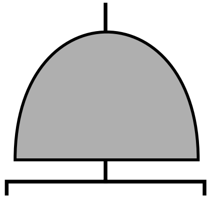
Basic Element



```
module BE_i
  fc, rc : clock;
  inform : [0..2] init 0;
  broken : [0..2] init 0;   // 0: up, 1: down, 2: repairing

  [fl!] broken=0 @ fc -> (inform=1) & (broken=1);
  [r??] broken=1      -> (broken=2) & (rc=γ);
  [up!] broken=2 @ rc -> (inform=2) &
                         (broken=0) & (fc=μ);

  [fi!!] inform=1 -> (inform=0);
  [ui!!] inform=2 -> (inform=0);
endmodule
```

Textual form of IOSA for the tool FIG

From RFT to IOSA



(Binary) AND gate

- if both inputs fail

  signal fault
- if one input repairs

  signal repair

```
module AND
  singalf: bool init false;
  signalu: bool init false;
  count: [0..2] init 0;

  [f1??] count=1 -> (count=2) & (signalf=true);
  [f1??] count=0 -> (count=1);
  [f2??] count=1 -> (count=2) & (signalf=true);
  [f2??] count=0 -> (count=1);

  [u1??] count=2 -> (count=1) & (signalu=true);
  [u1??] count=1 -> (count=0);
  [u2??] count=2 -> (count=1) & (signalu=true);
  [u2??] count=1 -> (count=0);

  [f!!] signalf & count=2  -> (signalf=false);
  [u!!] signalu & count!=2 -> (signalu=false);
endmodule
```

From RFT to IOSA

```
module OR
  signalf: bool init false;
  signalu: bool init false;
  count: [0..2] init 0;

  [f1??] count=0 -> (count'=1) & (signalf'=true);
  [f1??] count=1 -> (count'=2);
  [f2??] count=0 -> (count'=1) & (signalf'=true);
  [f2??] count=1 -> (count'=2);

  [u1??] count=2 -> (count'=1);
  [u1??] count=1 -> (count'=0) & (signalu'=true);
  [u2??] count=2 -> (count'=1);
  [u2??] count=1 -> (count'=0) & (signalu'=true);

  [f!!] signalf & count!=0 -> (signalf'=false);
  [u!!] signalu & count=0  -> (signalu'=false);
endmodule
```

```
module VOTING_3_1
  count: [0..3] init 0;
  inform: bool init false;

  [f0??] -> (count'=count+1) & (inform'=(count+1=2));
  [f1??] -> (count'=count+1) & (inform'=(count+1=2));
  [f2??] -> (count'=count+1) & (inform'=(count+1=2));

  [u0??] -> (count'=count-1) & (inform'=(count=2));
  [u1??] -> (count'=count-1) & (inform'=(count=2));
  [u2??] -> (count'=count-1) & (inform'=(count=2));

  [f!!] inform & count >= 2 -> (inform'=false);
  [u!!] inform & count < 2  -> (inform'=false);
endmodule
```

```
module PAND
  f1: bool init false;
  f2: bool init false;
  st: [0..4] init 0; // 0:up, 1:inform fail, 2:failed,
                     // 3:inform up, 4:unbreakable

  [_?] st=0 & f1 & !f0 -> (st'=4);

  [f0??] st=0 & !f0 & !f1 -> (f0'=true);
  [f0??] st=0 & !f0 & f1  -> (st'=1) & (f0'=true);
  [f0??] st!=0 & !f0      -> (f0'=true);
  [f0??] f0               ->;

  [f1??] st=0 & !f0 & !f1         -> (f1'=true);
  [f1??] st=0 & f0 & !f1          -> (st'=1) & (f1'=true);
  [f1??] st=3 & !f1               -> (st'=2) & (f1'=true);
  [f1??] (st==1|st==2|st=4) & !f1 -> (f1'=true);
  [f1??] f1                       ->;

  [u0??] st!=1 & f0 -> (f0'=false);
  [u0??] st=1 & f0  -> (st'=0) & (f0'=false);
  [u0??] !f0        ->;

  [u1??] (st=0|st=3) & f1 -> (f1'=false);
  [u1??] (st=1|st=4) & f1 -> (st'=0) & (f1'=false);
  [u1??] st=2 & f1        -> (st'=3) & (f1'=false);

  [f!!] st=1 -> (st'=2);
  [u!!] st=3 -> (st'=0);
endmodule
```

```
module RBOX
  broken[n]: bool init false;
  busy: bool init false;

  [fl_0?]  -> (broken[0]'=true);
  ...
  [fl_{n-1}?] -> (broken[n-1]'=true);

  [r0!!]  !busy & broken[0] -> (busy'=true);
  ...
  [r_{n-1}!!] !busy & broken[n-1]
               & !broken[n-2] & ... & !broken[0] -> (busy'=true);

  [up_0?] -> (broken[0]'=false) & (busy'=false);
  ...
  [up_{n-1}?] -> (broken[n-1]'=false) & (busy'=false);
endmodule
```

```
module SBE
  fc, dfc, rc : clock;
  inform : [0..2] init 0;
  active : bool init false;
  broken : [0..2] init 0;

  [e??] !active -> (active'=true) & (fc'=);
  [d??] active  -> (active'=false) & (dfc'= );

  [fl!] active & broken=0 @ fc   -> (inform'=1) & (broken'=1);
  [fl!] !active & broken=0 @ dfc -> (inform'=1) & (broken'=1);
  [r??]                          -> (broken'=2) & (rc'=);
  [up!] active & broken=2 @ rc   -> (inform'=2) & (broken'=0) & (fc'=);
  [up!] !active & broken=2 @ rc  -> (inform'=2) & (broken'=0) & (dfc'=);

  [f!!] inform=1 -> (inform'=0);
  [u!!] inform=2 -> (inform'=0);
endmodule
```

```
module MUX
  queue[n]: [0..3] init 0; // idle, requesting, reject, using
  avail: bool init true;
  broken: bool init false;
  enable: [0..2] init 0;

  [fl?] -> (broken'=true);
  [up?] -> (broken'=false);

  [e!!] enable=1 -> (enable'=0);
  [d!!] enable=2 -> (enable'=0);

  [rq_0??] queue[0]=0 & (broken | !avail)     -> (queue[0]'=2);
  [rq_0??] queue[0]=0 & !broken & avail       -> (queue[0]'=1);
  [asg_0!!] queue[0]=1 & !broken & avail      -> (queue[0]'=3) & (avail'=false);
  [rj_0!!] queue[0]=2                         -> (queue[0]'=1);
  [rel_0??] queue[0]=3                        -> (queue[0]'=0) & (avail'=true)
                                                 & (enable'=2);
  [acc_0??]                                   -> (enable'=1);
  ...
  [rq_{n-1}??] queue[n-1]=0 & (broken | !avail) -> (queue[n-1]'=2);
  [rq_{n-1}??] queue[n-1]=0 & !broken & avail   -> (queue[n-1]'=1);
  [asg_{n-1}!!] queue[n-1]=1 & queue[n-2]=0 & ...
                 & queue[0]=0 & !broken & avail  -> (queue[n-1]'=3) & (avail'=false);
  [rj_{n-1}!!] queue[n-1]=2                      -> (queue[n-1]'=1);
  [rel_{n-1}??] queue[n-1]=3                     -> (queue[n-1]'=0) & (avail'=true)
                                                   & (enable'=2);
  [acc_{n-1}??]                                  -> (enable'=1);
endmodule
```

```
module SPAREGATE
  state: [0..4] init 0; // on main, request, wait, on spare, broken
  inform: [0..2] init 0;
  release: [-n..n] init 0;
  idx: [1..n] init 1;

  [fl_0?] state=0        -> (state=1) & (idx=1);
  [up_0?] state=4        -> (state=0) & (inform=2);
  [up_0?] state=3 & idx=1 -> (state=0) & (idx=1) & (release=1);
  ...
  [up_0?] state=3 & idx=n -> (state=0) & (idx=1) & (release=n);

  [fl_1?] state=3 & idx=1 -> (release=1);
  ...
  [fl_n?] state=3 & idx=n -> (release=n);

  [rq_1!!] state=1 & idx=1 -> (state=2);
  ...
  [rq_n!!] state=1 & idx=n -> (state=2);

  [asg_1??] state=0 | state=1 | state=3 -> (release=1);
  [asg_1??] state=2 & idx=1             -> (release=-1) & (state=3);
  [asg_1??] state=4                     -> (release=-1) & (state=3)
                                           & (idx=1) & (inform=2);
  ...
  [asg_n??] state=0 | state=1 | state=3 -> (release=n);
  [asg_n??] state=2 & idx=n             -> (release=-n) & (state=3);
  [asg_n??] state=4                     -> (release=-n) & (state=3)
                                           & (idx=n) & (inform=2);

  [rj_1??] state=2 & idx=1 -> (idx=2) & (state=1);
  [rj_2??] state=2 & idx=2 -> (idx=3) & (state=1);
  ...
  [rj_n??] state=2 & idx=n -> (state=4) & (idx=1) & (inform=1);

  [rel_1!!] release=1 & !(state=3 & idx=1) -> (release= 0);
  [rel_1!!] release=1 & state=3 & idx=1    -> (release= 0) & (state=1) & (idx=1);
  ...

  [rel_n!!] release=n & !(state=3 & idx=n) -> (release=0);
  [rel_n!!] release=n & state=3 & idx=n    -> (release= 0) & (state=1) & (idx=1);

  [acc_1!!] release=-1 -> (release= 0);
  ...
  [acc_n!!] release=-n -> (release=0);

  [f!!] inform = 1 -> (inform=0);
  [u!!] inform = 2 -> (inform=0);
endmodule
```

CONICET

UNC

From RFT to IOSA+Urgency

Given a RFT $T = (V, i, si, l)$ the semantic of T is defined by

$$[\![T]\!] = ||_{v \in V} [\![v]\!]$$

where

$$[\![v]\!] = \begin{cases} [\![l(v)]\!](\mathtt{fl}_v, \mathtt{up}_v, \mathtt{f}_v, \mathtt{u}_v, \mathtt{r}_v) & \text{if } l(v) = (\text{be}, 0, \mu, \gamma) \\ [\![l(v)]\!](\mathtt{f}_v, \mathtt{u}_v, \mathtt{f}_{i(v)[0]}, \mathtt{u}_{i(v)[0]}, \ldots, \mathtt{f}_{i(v)[n-1]}, \mathtt{u}_{i(v)[n-1]}) & \text{if } l(v) \in \{(\text{and}, n), (\text{or}, n)\} \\ [\![l(v)]\!](\mathtt{f}_v, \mathtt{u}_v, \mathtt{f}_{i(v)[0]}, \mathtt{u}_{i(v)[0]}, \mathtt{f}_{i(v)[1]}, \mathtt{u}_{i(v)[1]}) & \text{if } l(v) = (\text{pand}, 2) \\ [\![l(v)]\!](\mathtt{fl}_{i(v)[0]}, \mathtt{up}_{i(v)[0]}, \mathtt{r}_{i(v)[0]}, \ldots, \mathtt{fl}_{i(v)[n-1]}, \mathtt{up}_{i(v)[n-1]}, \mathtt{r}_{i(v)[n-1]}) & \text{if } l(v) = (\text{rbox}, n) \\ [\![l(v)]\!](\mathtt{fl}_v, \mathtt{up}_v, \mathtt{f}_v, \mathtt{u}_v, \mathtt{r}_v, \mathtt{e}_v, \mathtt{d}_v, \mathtt{rq}_{(si(v)[0],v)}, \mathtt{asg}_{(v,si(v)[0])}, & \\ \qquad \mathtt{rel}_{(si(v)[0],v)}, \mathtt{acc}_{(si(v)[0],v)}, \mathtt{rj}_{(v,si(v)[0])}, .., \mathtt{rj}_{(v,si(v)[n-1])}) & \text{if } l(v) = (\text{sbe}, n, \mu, \nu, \gamma) \\ [\![l(v)]\!](\mathtt{f}_v, \mathtt{u}_v, \mathtt{fl}_{i(v)[0]}, \mathtt{up}_{i(v)[0]}, \mathtt{fl}_{i(v)[1]}, \mathtt{up}_{i(v)[1]}, \mathtt{rq}_{(v,i(v)[1])}, \mathtt{asg}_{(i(v)[1],v)}, & \\ \qquad \mathtt{acc}_{(v,i(v)[1])}, \mathtt{rj}_{(i(v)[1],v)}, \mathtt{rel}_{(v,i(v)[1])}, \ldots, \mathtt{rel}_{(v,i(v)[n-1])}) & \text{if } l(v) = (\text{sg}, n) \end{cases}$$

From RFT to IOSA+Urgency

Given a RFT $T = (V, i, si, l)$ the semantic of T is defined by

$$[\![T]\!] = ||_{v \in V} [\![v]\!]$$

where

The encodings given before with proper relabeling

$$[\![v]\!] = \begin{cases} [\![l(v)]\!](\texttt{fl}_v, \texttt{up}_v, \texttt{f}_v, \texttt{u}_v, \texttt{r}_v) & \text{if } l(v) = (\text{be}, 0, \mu, \gamma) \\ [\![l(v)]\!](\texttt{f}_v, \texttt{u}_v, \texttt{f}_{i(v)[0]}, \texttt{u}_{i(v)[0]}, \ldots, \texttt{f}_{i(v)[n-1]}, \texttt{u}_{i(v)[n-1]}) & \text{if } l(v) \in \{(\text{and}, n), (\text{or}, n)\} \\ [\![l(v)]\!](\texttt{f}_v, \texttt{u}_v, \texttt{f}_{i(v)[0]}, \texttt{u}_{i(v)[0]}, \texttt{f}_{i(v)[1]}, \texttt{u}_{i(v)[1]}) & \text{if } l(v) = (\text{pand}, 2) \\ [\![l(v)]\!](\texttt{fl}_{i(v)[0]}, \texttt{up}_{i(v)[0]}, \texttt{r}_{i(v)[0]}, \ldots, \texttt{fl}_{i(v)[n-1]}, \texttt{up}_{i(v)[n-1]}, \texttt{r}_{i(v)[n-1]}) & \text{if } l(v) = (\text{rbox}, n) \\ [\![l(v)]\!](\texttt{fl}_v, \texttt{up}_v, \texttt{f}_v, \texttt{u}_v, \texttt{r}_v, \texttt{e}_v, \texttt{d}_v, \texttt{rq}_{(si(v)[0],v)}, \texttt{asg}_{(v,si(v)[0])}, & \\ \qquad \texttt{rel}_{(si(v)[0],v)}, \texttt{acc}_{(si(v)[0],v)}, \texttt{rj}_{(v,si(v)[0])}, .., \texttt{rj}_{(v,si(v)[n-1])}) & \text{if } l(v) = (\text{sbe}, n, \mu, \nu, \gamma) \\ [\![l(v)]\!](\texttt{f}_v, \texttt{u}_v, \texttt{fl}_{i(v)[0]}, \texttt{up}_{i(v)[0]}, \texttt{fl}_{i(v)[1]}, \texttt{up}_{i(v)[1]}, \texttt{rq}_{(v,i(v)[1])}, \texttt{asg}_{(i(v)[1],v)}, & \\ \qquad \texttt{acc}_{(v,i(v)[1])}, \texttt{rj}_{(i(v)[1],v)}, \texttt{rel}_{(v,i(v)[1])}, \ldots, \texttt{rel}_{(v,i(v)[n-1])}) & \text{if } l(v) = (\text{sg}, n) \end{cases}$$

From RFT to IOSA+Urgency

Given a RFT $T = (V, i, si, l)$ the semantic of T is defined by

$$[\![T]\!] = ||_{v \in V} [\![v]\!]$$

where

$\Big\{ [\![l(v)]\!](\texttt{fl}_v, \texttt{up}_v, \texttt{f}_v$ if $l(v) = (\text{be}, 0, \mu, \gamma)$

if $l(v) \in \{(\text{and}, n), (\text{or}, n)\}$

$(v)[0], \texttt{u}_{i(v)[0]}, \texttt{f}_{i(v)[1]}, \texttt{u}_{i(v)[1]})$ if $l(v) = (\text{pand}, 2)$

$\texttt{up}_{i(v)[0]}, \texttt{r}_{i(v)[0]}, \ldots, \texttt{fl}_{i(v)[n-1]}, \texttt{up}_{i(v)[n-1]}, \texttt{r}_{i(v)[n-1]})$ if $l(v) = (\text{rbox}, n)$

$\texttt{f}_v, \texttt{u}_v, \texttt{r}_v, \texttt{e}_v, \texttt{d}_v, \texttt{rq}_{(si(v)[0],v)}, \texttt{asg}_{(v,si(v)[0])},$

$[0],v), \texttt{acc}_{(si(v)[0],v)}, \texttt{rj}_{(v,si(v)[0])}, .., \texttt{rj}_{(v,si(v)[n-1])})$ if $l(v) = (\text{sbe}, n, \mu, \nu, \gamma)$

$_{i(v)[0]}, \texttt{up}_{i(v)[0]}, \texttt{fl}_{i(v)[1]}, \texttt{up}_{i(v)[1]}, \texttt{rq}_{(v,i(v)[1])}, \texttt{asg}_{(i(v)[1],v)},$

$)[1]), \texttt{rj}_{(i(v)[1],v)}, \texttt{rel}_{(v,i(v)[1])}, \ldots, \texttt{rel}_{(v,i(v)[n-1])})$ if $l(v) = (\text{sg}, n)$

From RFT to IOSA+Urgency

Given a RFT $T = (V, i, si, l)$ the semantic of T is defined by

$$[\![T]\!] = ||_{v \in V} [\![v]\!]$$

where

It satisfies the sufficient conditions that guarantee confluence. Hence, it is weakly deterministic!

$\left\{ [\![l(v)]\!](\mathtt{fl}_v, \mathtt{up}_v, \mathtt{f}_v \right.$ if $l(v) = (\mathsf{be}, 0, \mu, \gamma)$

$(v)[$ if $l(v) \in \{(\mathsf{and}, n), (\mathsf{or}, n)\}$

$(v)[0], \mathtt{u}_{i(v)[0]}, \mathtt{f}_{i(v)[1]}, \mathtt{u}_{i(v)[1]})$ if $l(v) = (\mathsf{pand}, 2)$

$\mathtt{up}_{i(v)[0]}, \mathtt{r}_{i(v)[0]}, \ldots, \mathtt{fl}_{i(v)[n-1]}, \mathtt{up}_{i(v)[n-1]}, \mathtt{r}_{i(v)[n-1]})$ if $l(v) = (\mathsf{rbox}, n)$

$\mathtt{f}_v, \mathtt{u}_v, \mathtt{r}_v, \mathtt{e}_v, \mathtt{d}_v, \mathtt{rq}_{(si(v)[0],v)}, \mathtt{asg}_{(v,si(v)[0])},$

$[0],v), \mathtt{acc}_{(si(v)[0],v)}, \mathtt{rj}_{(v,si(v)[0])}, .., \mathtt{rj}_{(v,si(v)[n-1])})$ if $l(v) = (\mathsf{sbe}, n, \mu, \nu, \gamma)$

$_{i(v)[0]}, \mathtt{up}_{i(v)[0]}, \mathtt{fl}_{i(v)[1]}, \mathtt{up}_{i(v)[1]}, \mathtt{rq}_{(v,i(v)[1])}, \mathtt{asg}_{(i(v)[1],v)},$

$)[1]), \mathtt{rj}_{(i(v)[1],v)}, \mathtt{rel}_{(v,i(v)[1])}, \ldots, \mathtt{rel}_{(v,i(v)[n-1])})$ if $l(v) = (\mathsf{sg}, n)$

From RFT to IOSA+Urgency

Given a RFT $T = (V, i, si, l)$ the semantic of T is defined by

$$[\![T]\!] = ||_{v \in V} [\![v]\!]$$

where

$\Big\{ [\![l(v)]\!](\mathtt{fl}_v, \mathtt{up}_v, \mathtt{f}_v$ if $l(v) = (\mathsf{be}, 0, \mu, \gamma)$

$(v)[$ if $l(v) \in \{(\mathsf{and}, n), (\mathsf{or}, n)\}$

$(v)[0], \mathtt{u}_{i(v)[0]}, \mathtt{f}_{i(v)[1]}, \mathtt{u}_{i(v)[1]})$ if $l(v) = (\mathsf{pand}, 2)$

$\mathtt{up}_{i(v)[0]}, \mathtt{r}_{i(v)[0]}, \ldots, \mathtt{fl}_{i(v)[n-1]}, \mathtt{up}_{i(v)[n-1]}, \mathtt{r}_{i(v)[n-1]})$ if $l($

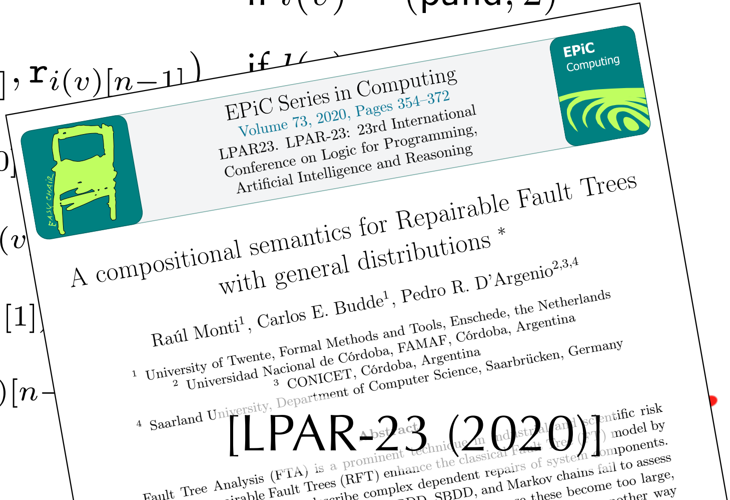
$\mathtt{f}_v, \mathtt{u}_v, \mathtt{r}_v, \mathtt{e}_v, \mathtt{d}_v, \mathtt{rq}_{(si(v)[0],v)}, \mathtt{asg}_{(v,si(v)[0]}$

$[0],v), \mathtt{acc}_{(si(v)[0],v)}, \mathtt{rj}_{(v,si(v)[0])}, .., \mathtt{rj}_{(v,si(v}$

$\mathtt{l}_{i(v)[0]}, \mathtt{up}_{i(v)[0]}, \mathtt{fl}_{i(v)[1]}, \mathtt{up}_{i(v)[1]}, \mathtt{rq}_{(v,i(v)[1]}$

$)[1]), \mathtt{rj}_{(i(v)[1],v)}, \mathtt{rel}_{(v,i(v)[1])}, \ldots, \mathtt{rel}_{(v,i(v)[n-}$

It satisfies the sufficient conditions
that guarantee confluence.
Hence, it is weakly deterministic!



EPiC Series in Computing
Volume 73, 2020, Pages 354–372
LPAR23. LPAR-23: 23rd International
Conference on Logic for Programming,
Artificial Intelligence and Reasoning

A compositional semantics for Repairable Fault Trees with general distributions *

Raúl Monti[1], Carlos E. Budde[1], Pedro R. D'Argenio[2,3,4]

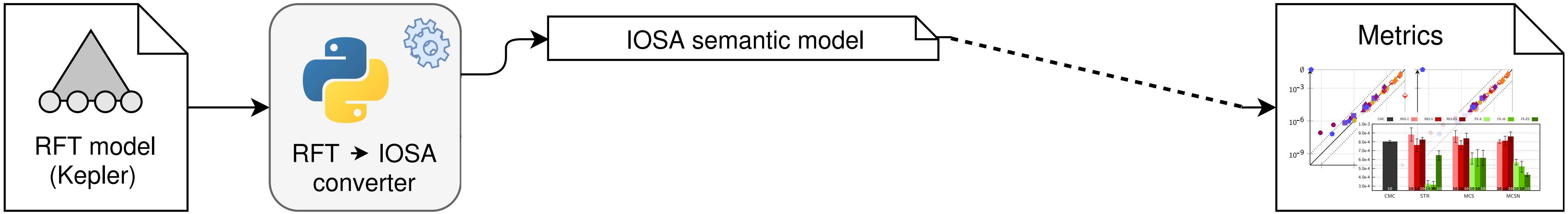
[1] University of Twente, Formal Methods and Tools, Enschede, the Netherlands
[2] Universidad Nacional de Córdoba, FAMAF, Córdoba, Argentina
[3] CONICET, Córdoba, Argentina
[4] Saarland University, Department of Computer Science, Saarbrücken, Germany

Abstract

Fault Tree Analysis (FTA) is a prominent technique in industrial and scientific risk
airable Fault Trees (RFT) enhance the classical Fault Tree (FT) model by
scribe complex dependent repairs of system components.
BDD, SBDD, and Markov chains fail to assess
se these become too large,
other way

[LPAR-23 (2020)]

Building the Tool Chain



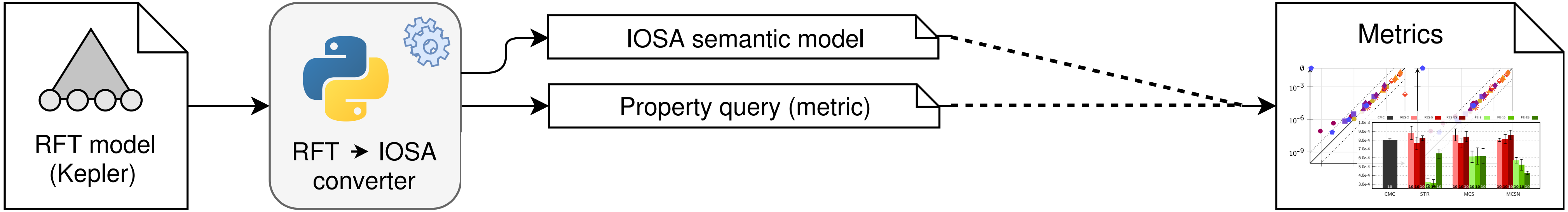
RFT model (Kepler)

RFT → IOSA converter

IOSA semantic model

Metrics

Building the Tool Chain



RFT model (Kepler)

RFT ➔ IOSA converter

IOSA semantic model

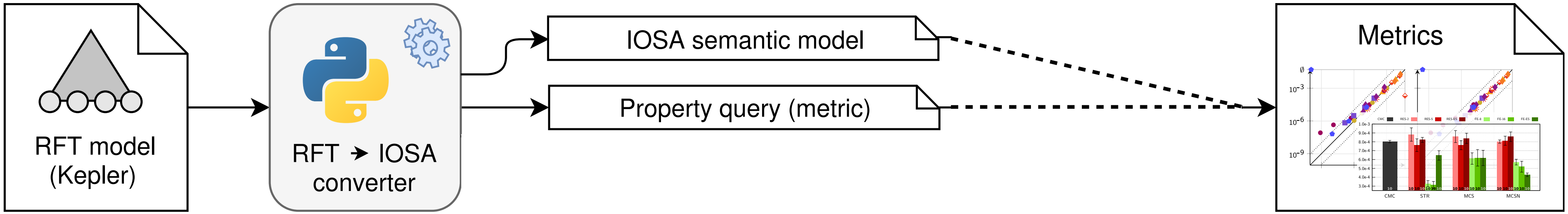
Property query (metric)

Metrics

Reliability: $\mathbb{P}(\Box_{\leq T}\neg\mathrm{TLE})$ (transient)

Availability: $\mathbb{E}(\neg\mathrm{TLE})$ (steady-state)

Building the Tool Chain



RFT model (Kepler)

RFT ➔ IOSA converter

IOSA semantic model

Property query (metric)

Metrics

Reliability:	$1 - \mathbb{P}(\Diamond_{\leq T}\mathrm{TLE})$	(transient)
Availability:	$\mathbb{E}(\neg\mathrm{TLE})$	(steady-state)

Monte Carlo Simulation

Prob (*unsafe* U *fail*) ?

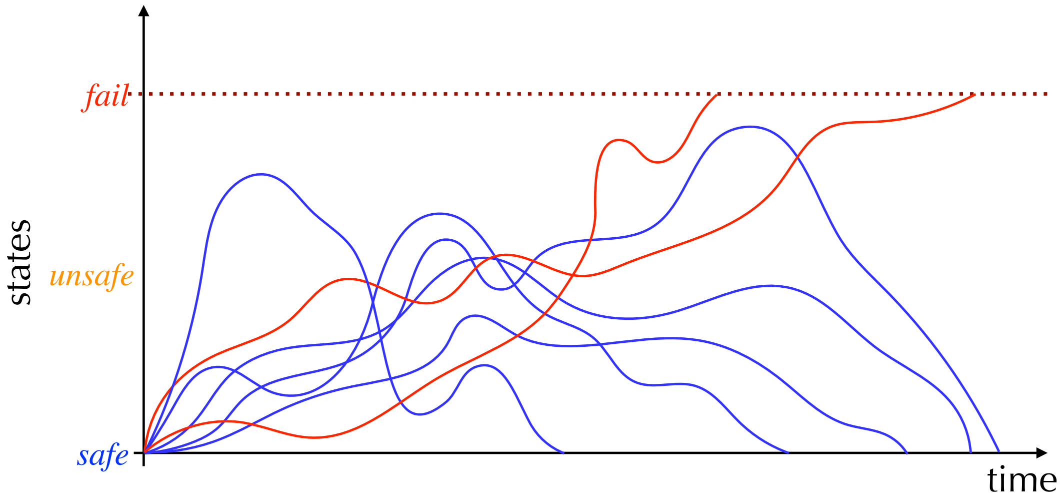
fail
states
unsafe
safe
time

Monte Carlo Simulation

#✘ = 2

#total = 7

$$\mathrm{Prob}\,(\,\mathit{unsafe}\ \mathsf{U}\ \mathit{fail}\,) \approx \hat{p} = \frac{\#✘}{\#\mathrm{total}}$$



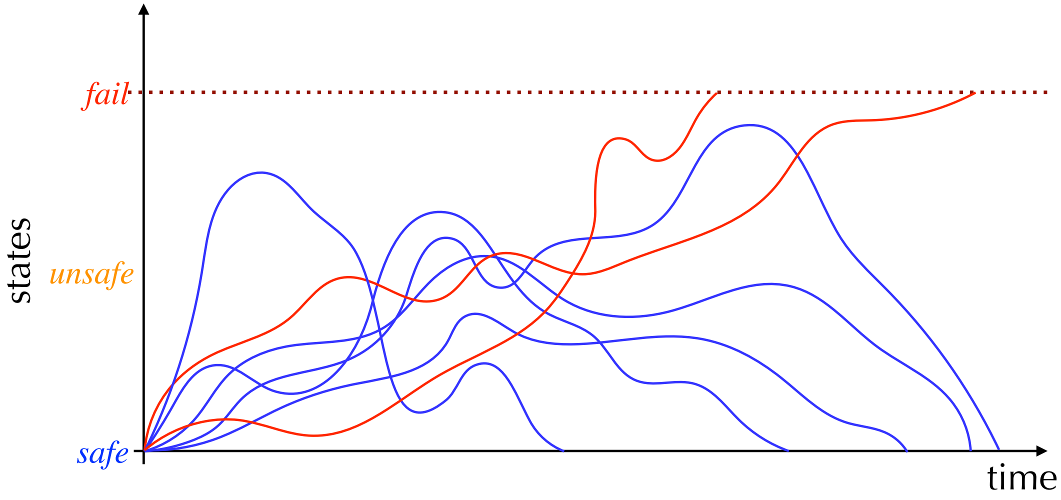
Monte Carlo Simulation

Highly Reliable

#✗ = 2

#total = 7

$$\text{Prob}\,(\,\textit{unsafe}\ \mathsf{U}\ \textit{fail}\,) \approx \hat{p} = \frac{\#✗}{\#\text{total}}$$



Monte Carlo Simulation

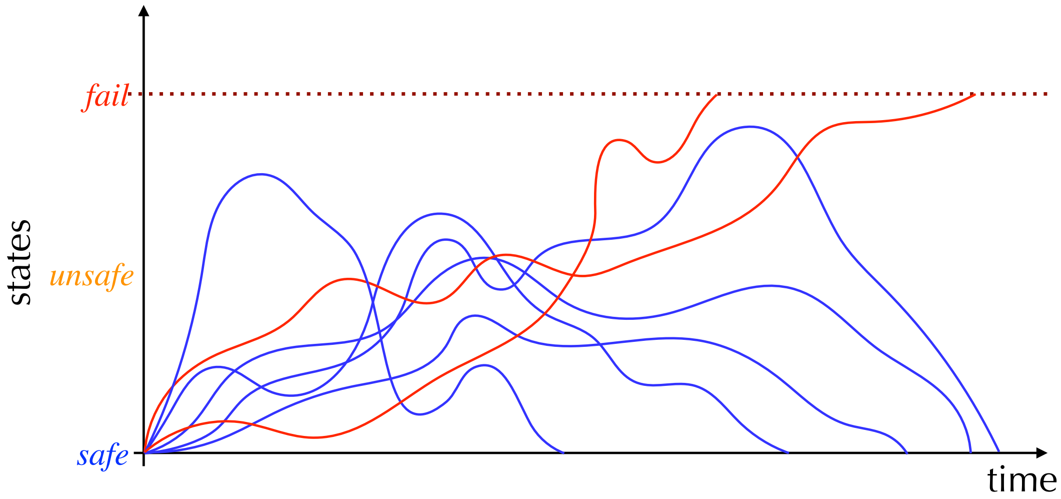
Highly Reliable

Too small

#✗ = 2

#total = 7

$$\text{Prob}\ (\ \textit{unsafe}\ \mathbf{U}\ \textit{fail}\ ) \approx \hat{p} = \frac{\#\text{✗}}{\#\text{total}}$$



Monte Carlo Simulation

Highly Reliable

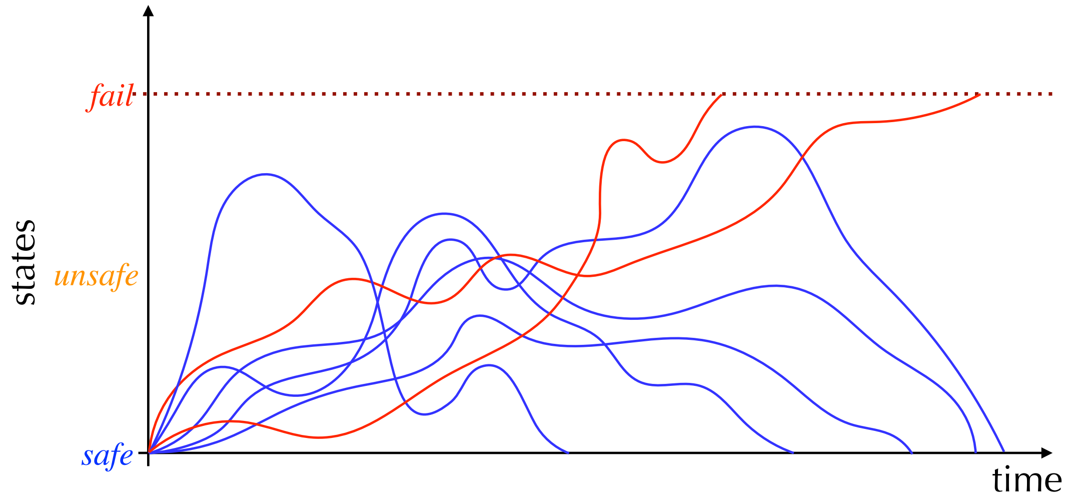
Too small

Too few

#✗ = 2

#total = 7

$$\text{Prob}\,(\,\textit{unsafe}\ \mathsf{U}\ \textit{fail}\,) \approx \hat{p} = \frac{\#\text{✗}}{\#\text{total}}$$



Monte Carlo Simulation

Highly Reliable

Too small

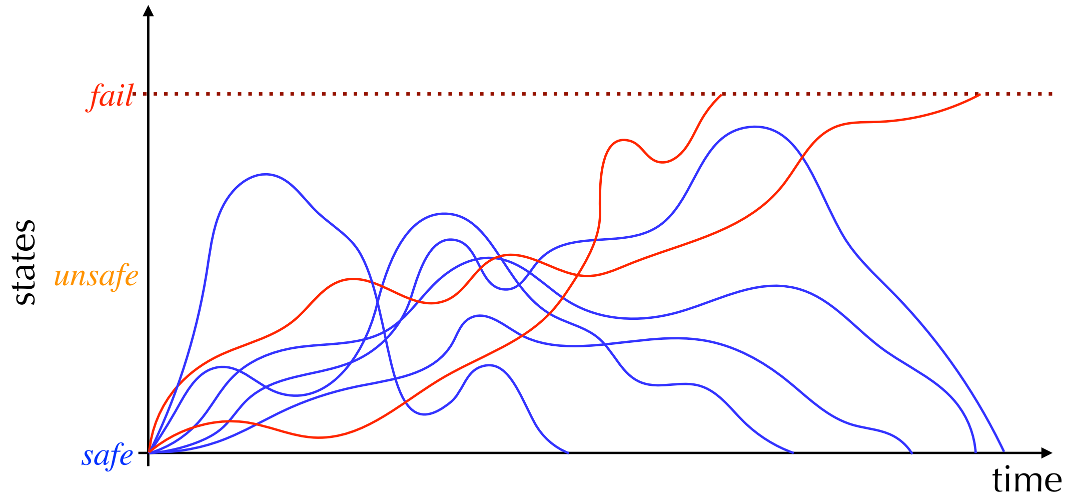
Too few

$\#✗ = 2$

$\#\text{total} = 7$

$$\text{Prob}\,(\,\textit{unsafe}\ \mathsf{U}\ \textit{fail}\,) \approx \hat{p} = \frac{\#✗}{\#\text{total}}$$

Needs to be huge



Monte Carlo Simulation

Highly Reliable

Too small

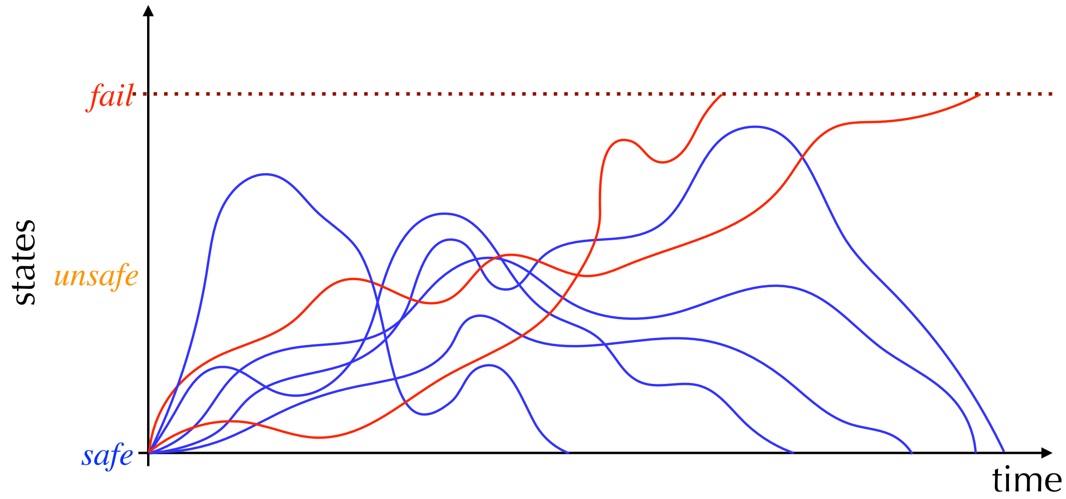
Too few

Needs to be huge

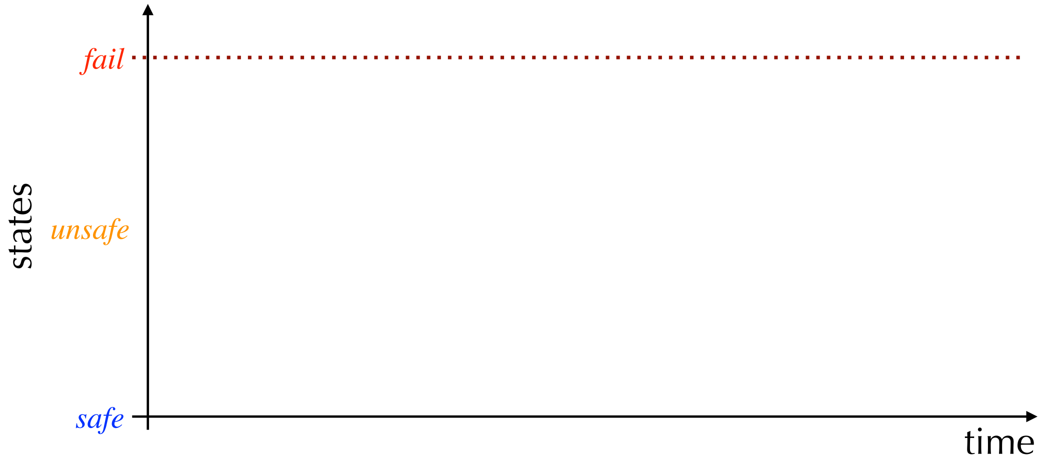
#✗ = 2

#total = 7

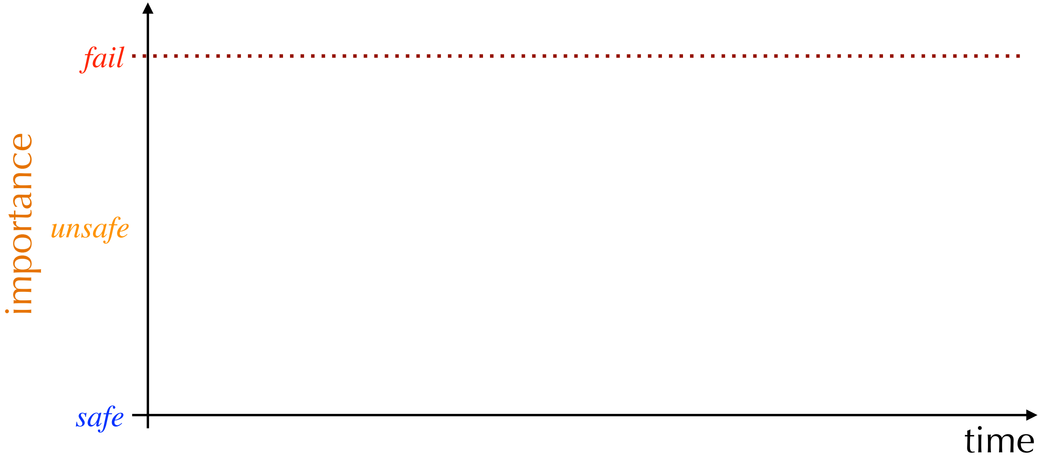
$$\text{Prob}\,(\,\mathit{unsafe}\ \mathsf{U}\ \mathit{fail}\,) \approx \hat{p} = \frac{\#\text{✗}}{\#\text{total}}$$



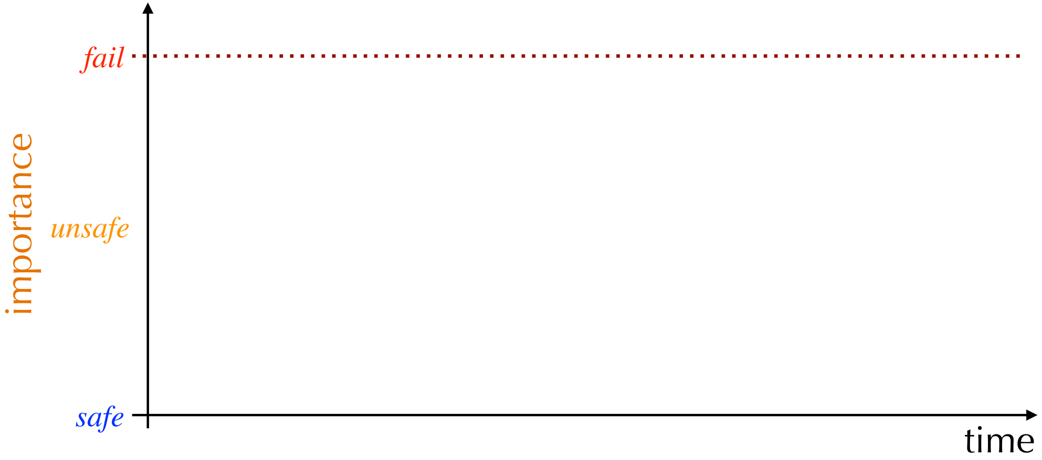
Rare event simulation through Importance Splitting



Rare event simulation through Importance Splitting



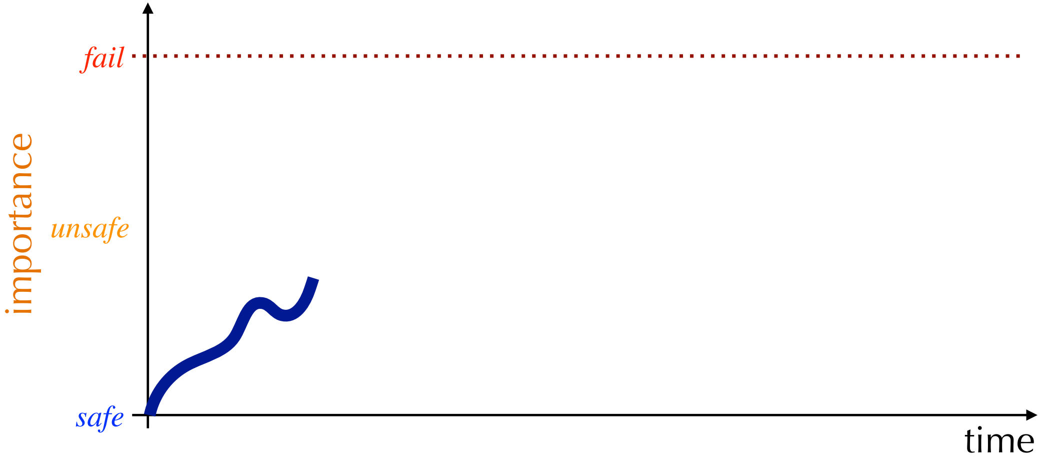
Rare event simulation through Importance Splitting



rare event

Ideally indicates the "proximity" to the rare event

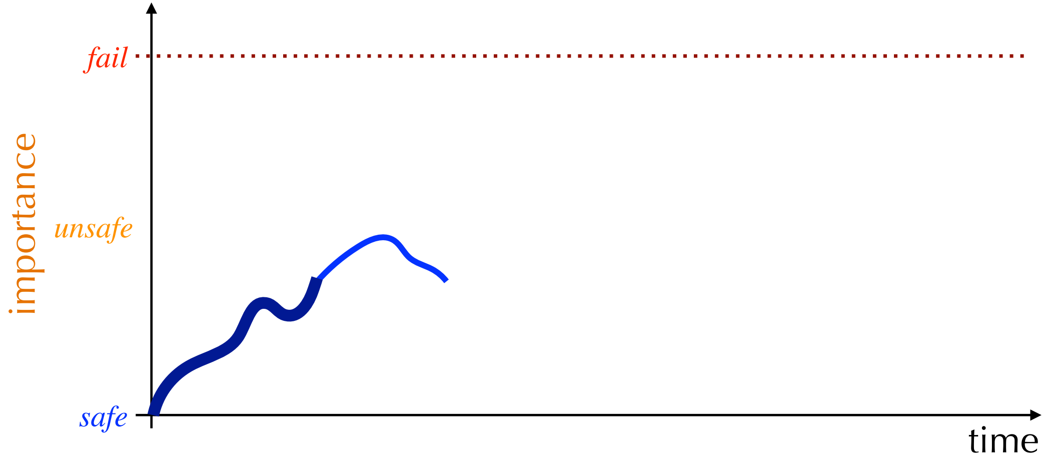
Rare event simulation through Importance Splitting



rare event

Ideally indicates the "proximity" to the rare event

Rare event simulation through Importance Splitting

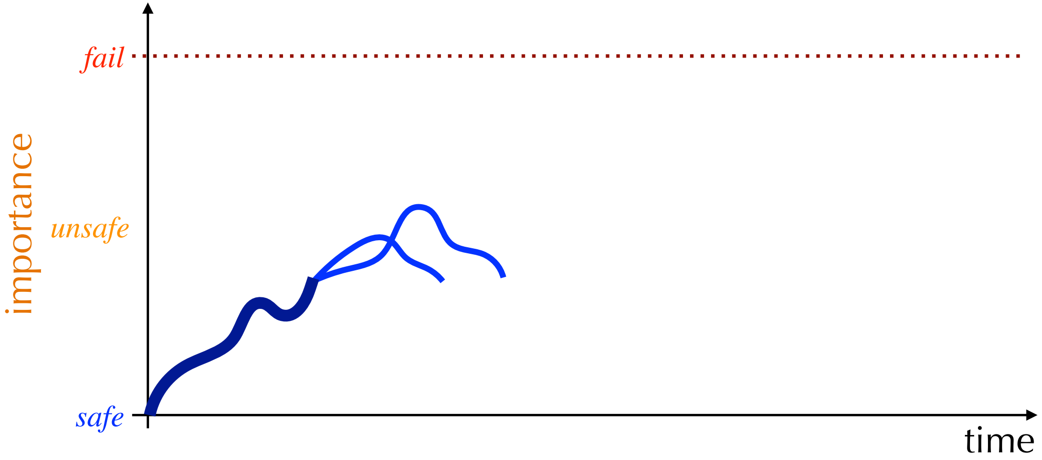


rare event

Ideally indicates the “proximity” to the rare event

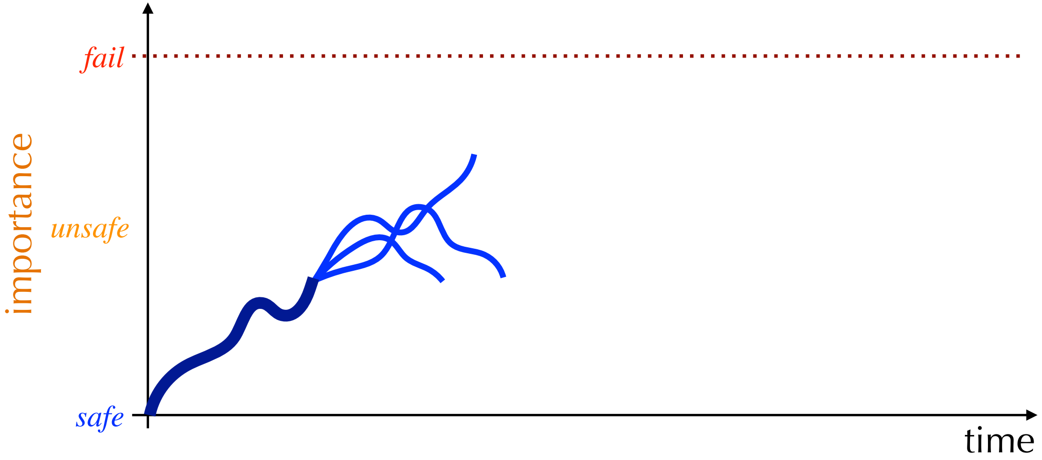
Rare event simulation through Importance Splitting

Ideally indicates the "proximity" to the rare event



rare event

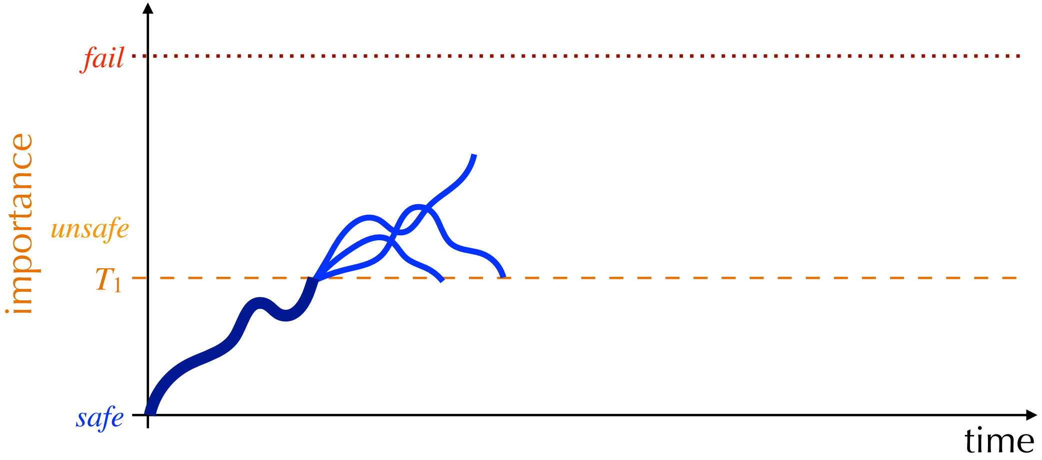
Rare event simulation through Importance Splitting



rare event

Ideally indicates the "proximity" to the rare event

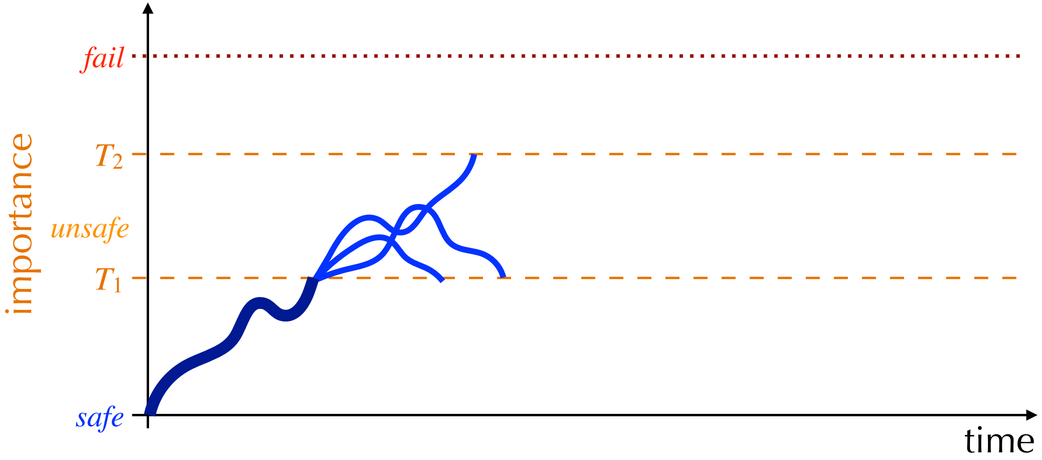
Rare event simulation through Importance Splitting



rare event

Ideally indicates the "proximity" to the rare event

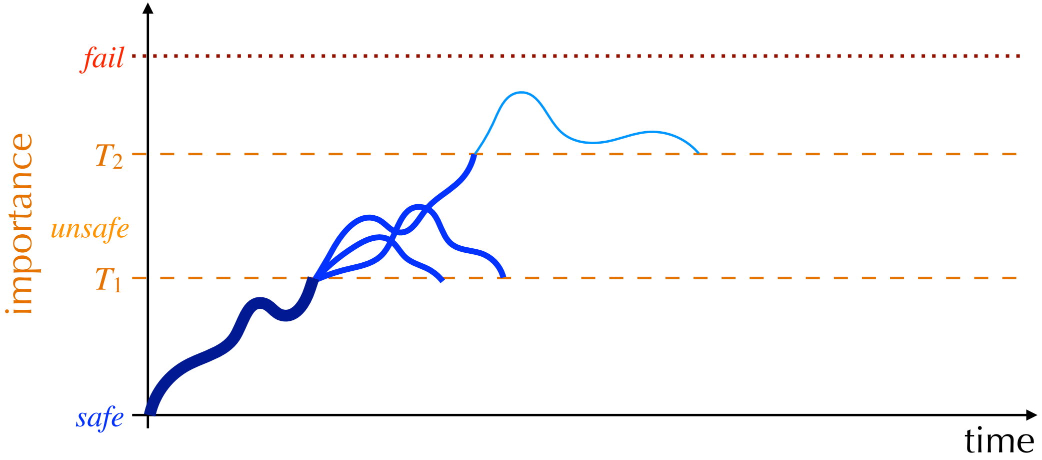
Rare event simulation through Importance Splitting



rare event

Ideally indicates the “proximity” to the rare event

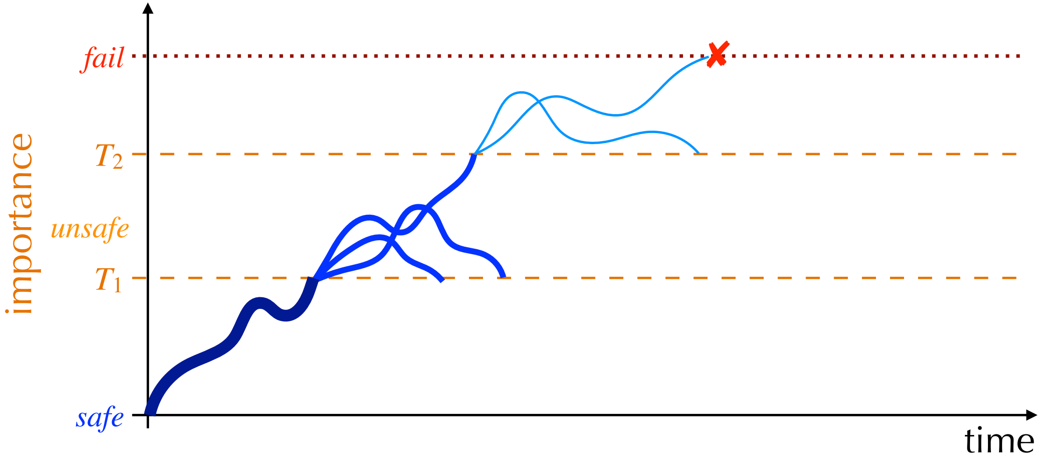
Rare event simulation through Importance Splitting



rare event

Ideally indicates the “proximity” to the rare event

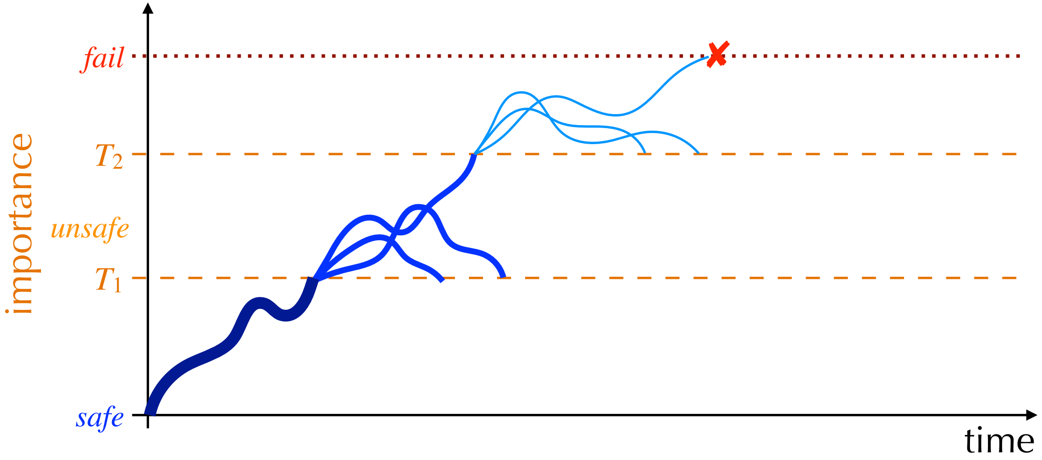
Rare event simulation through Importance Splitting



rare event

Ideally indicates the “proximity” to the rare event

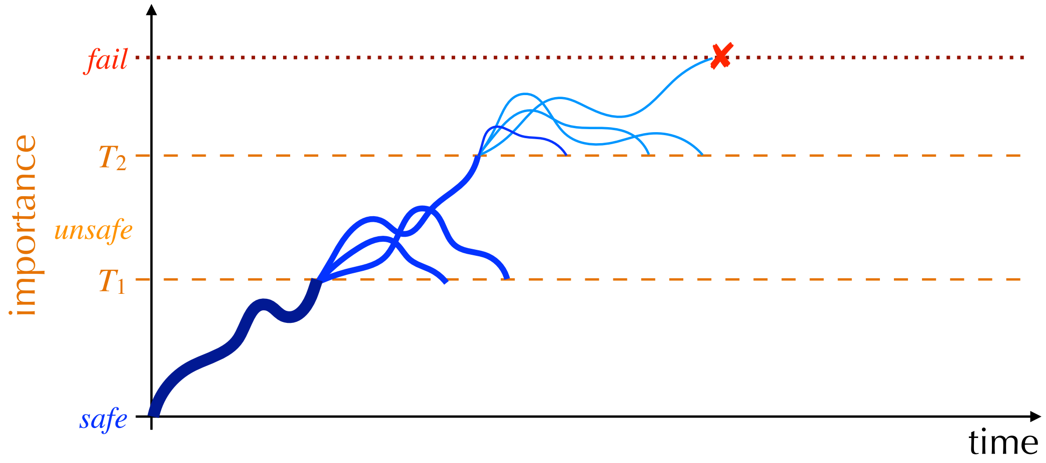
Rare event simulation through Importance Splitting



rare event

Ideally indicates the "proximity" to the rare event

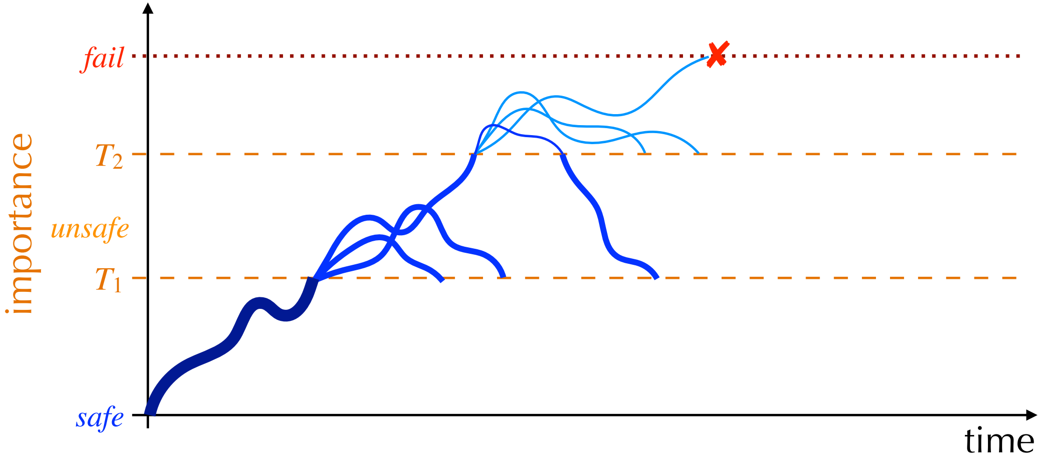
Rare event simulation through Importance Splitting



rare event

Ideally indicates the "proximity" to the rare event

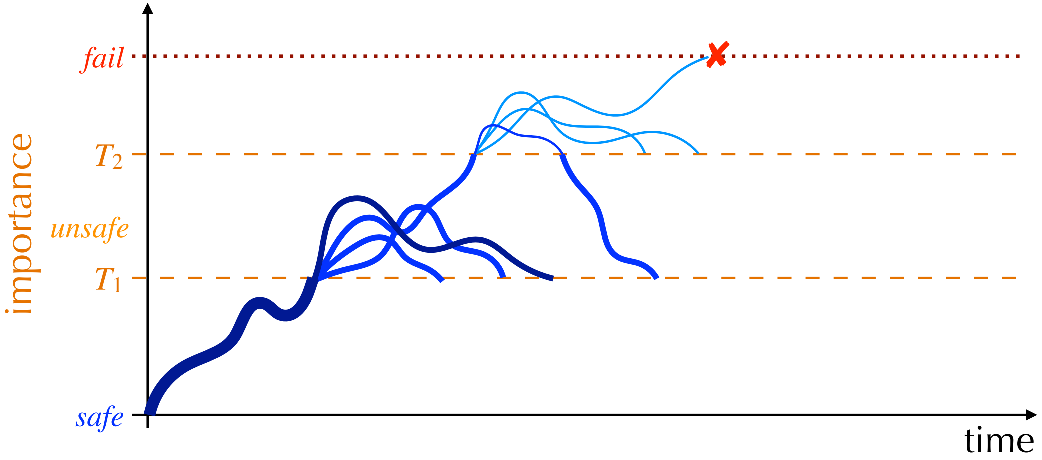
Rare event simulation through Importance Splitting



rare event

Ideally indicates the "proximity" to the rare event

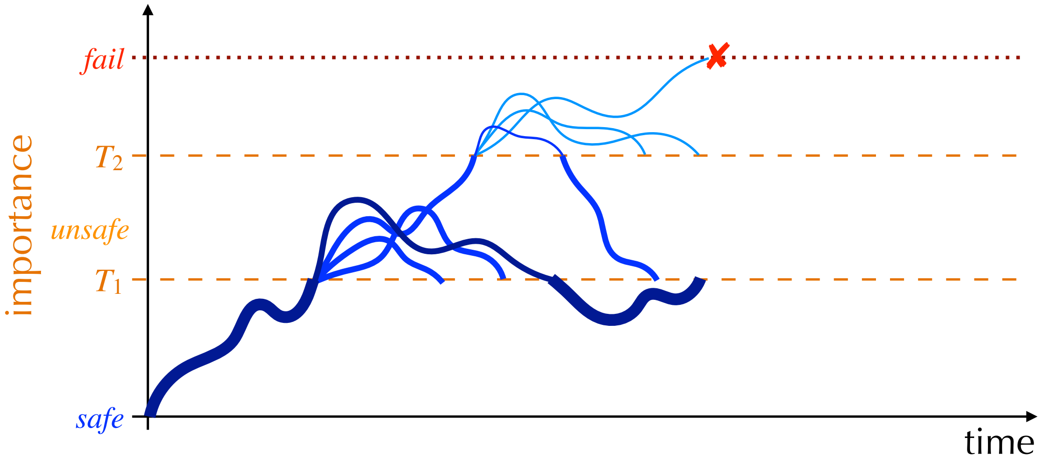
Rare event simulation through Importance Splitting



rare event

Ideally indicates the "proximity" to the rare event

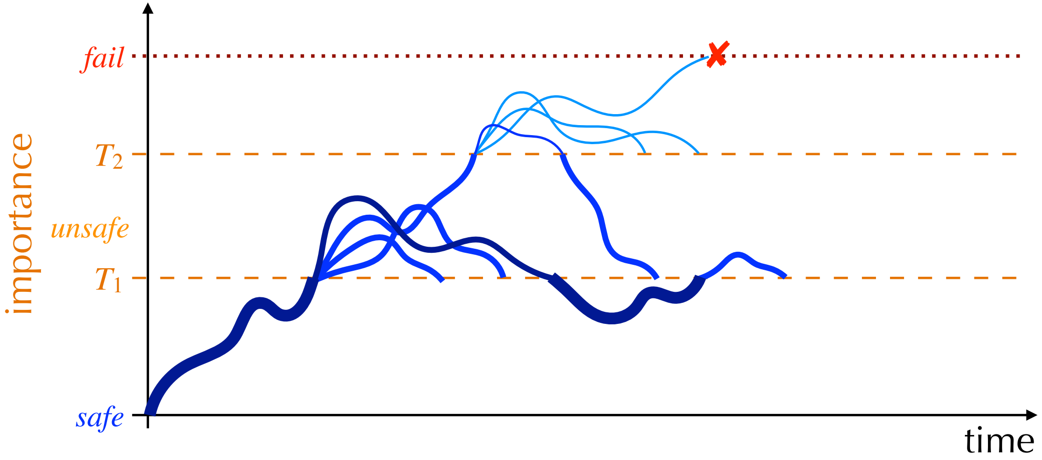
Rare event simulation through Importance Splitting



rare event

Ideally indicates the “proximity” to the rare event

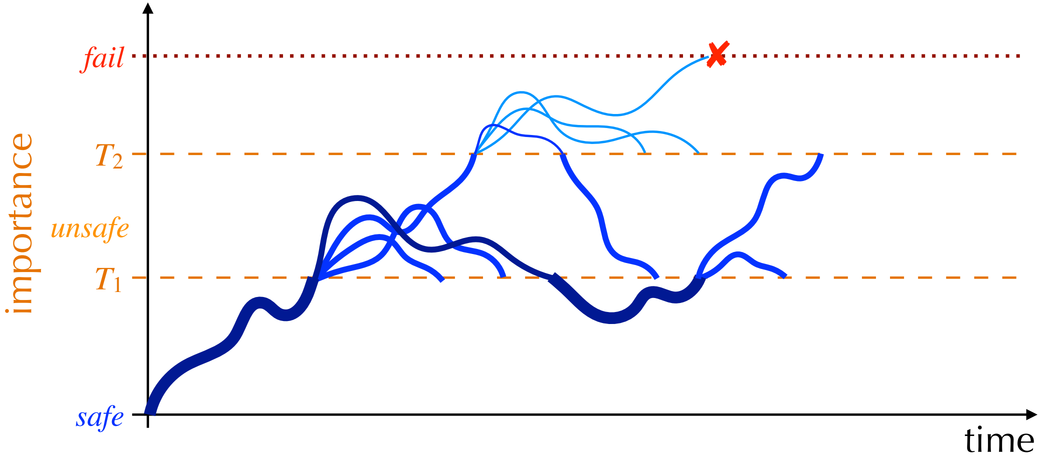
Rare event simulation through Importance Splitting



rare event

Ideally indicates the "proximity" to the rare event

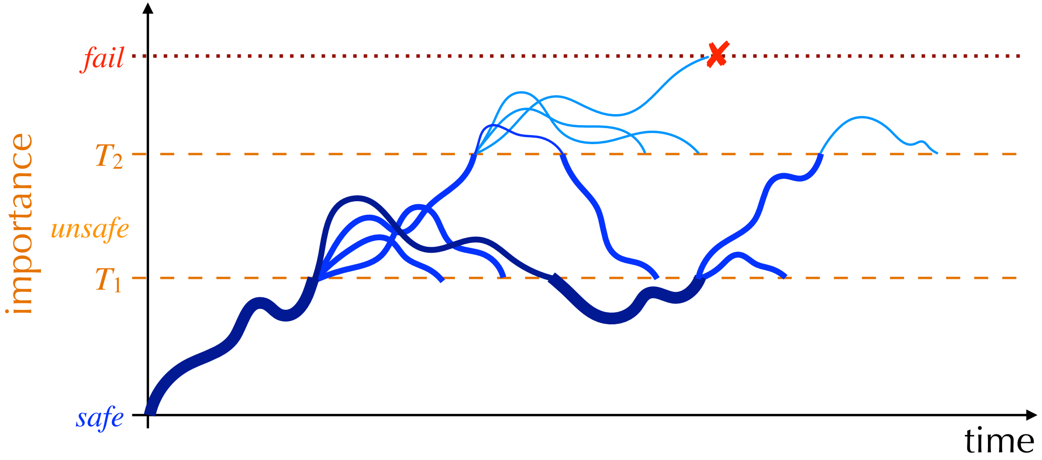
Rare event simulation through Importance Splitting



rare event

Ideally indicates the “proximity” to the rare event

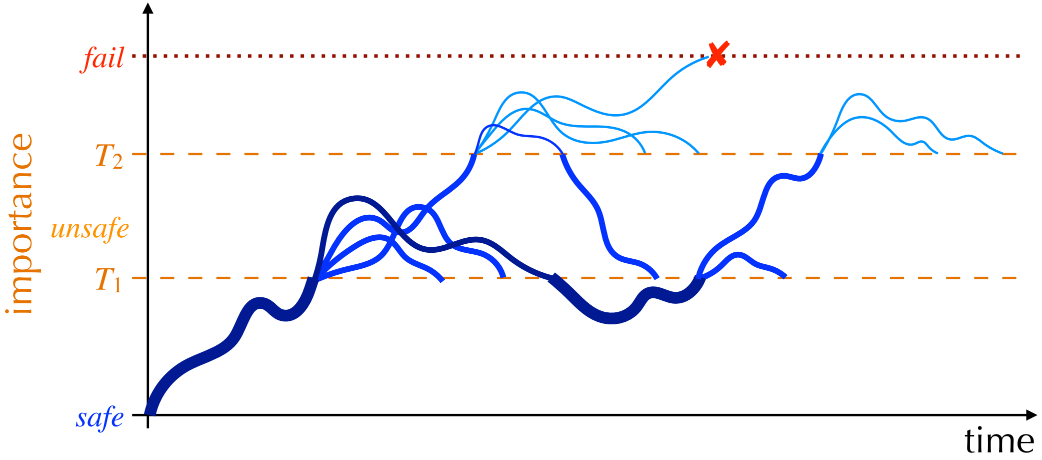
Rare event simulation through Importance Splitting



rare event

Ideally indicates the “proximity” to the rare event

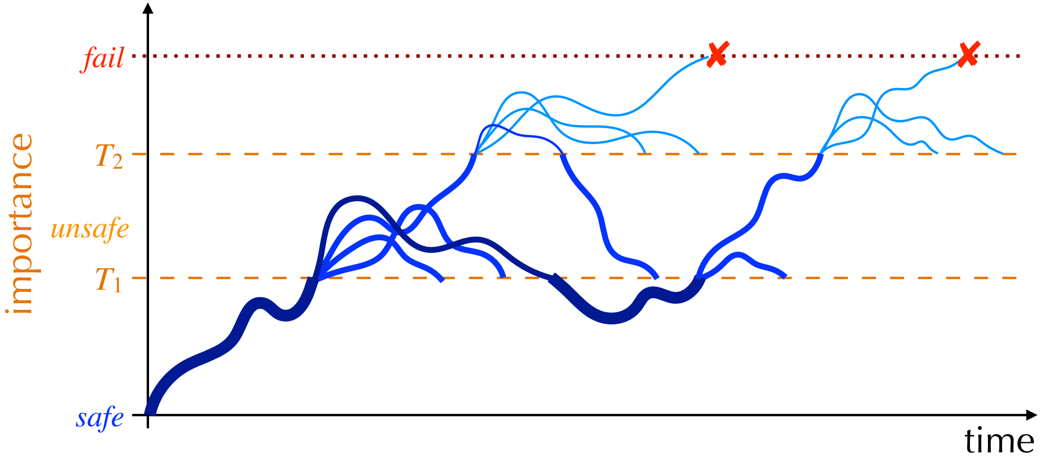
Rare event simulation through Importance Splitting



rare event

Ideally indicates the “proximity” to the rare event

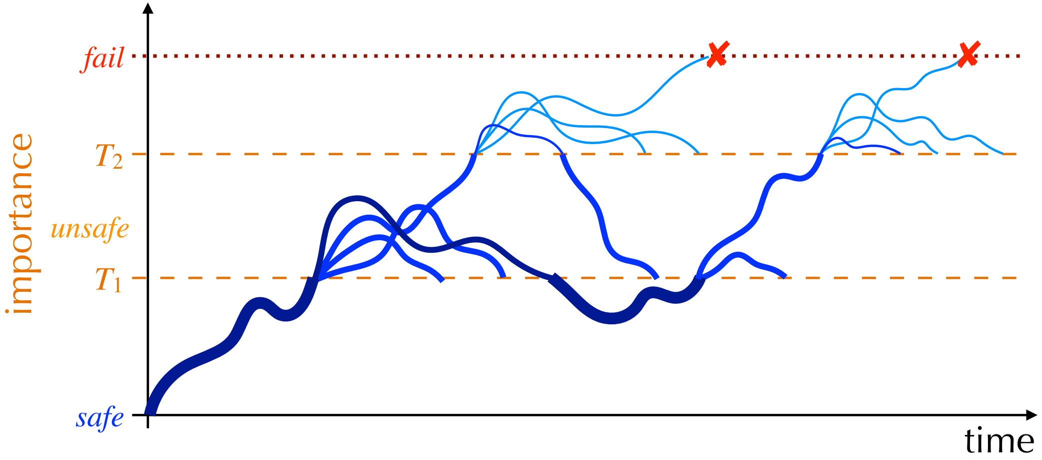
Rare event simulation through Importance Splitting



rare event

Ideally indicates the “proximity” to the rare event

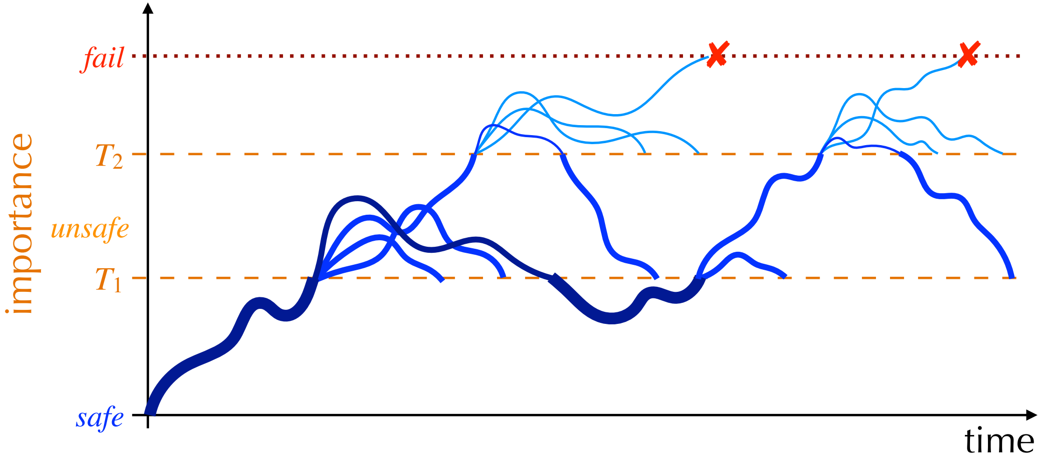
Rare event simulation through Importance Splitting



rare event

Ideally indicates the "proximity" to the rare event

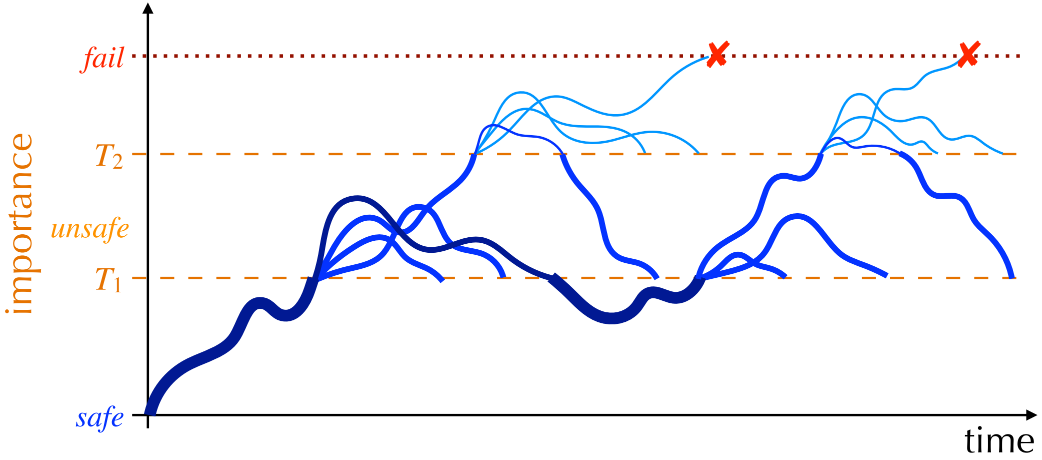
Rare event simulation through Importance Splitting



rare event

Ideally indicates the “proximity” to the rare event

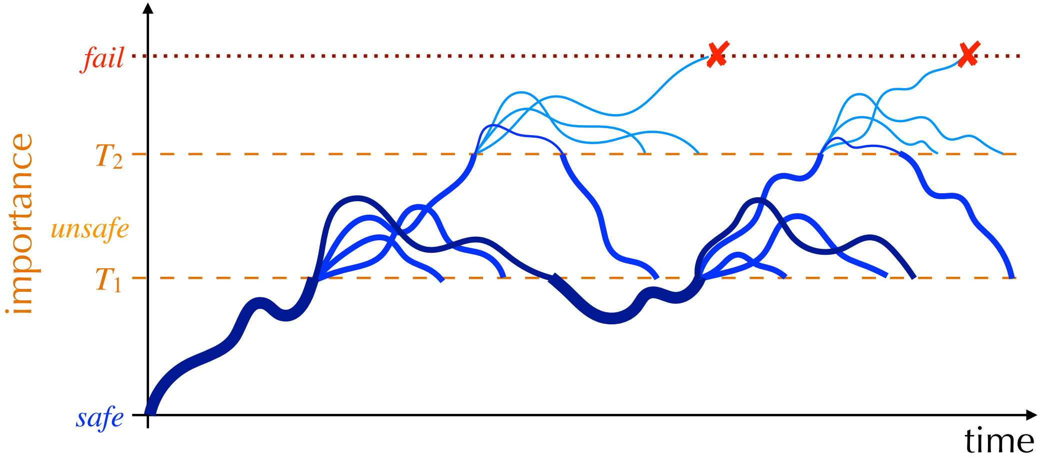
Rare event simulation through Importance Splitting



rare event

Ideally indicates the “proximity” to the rare event

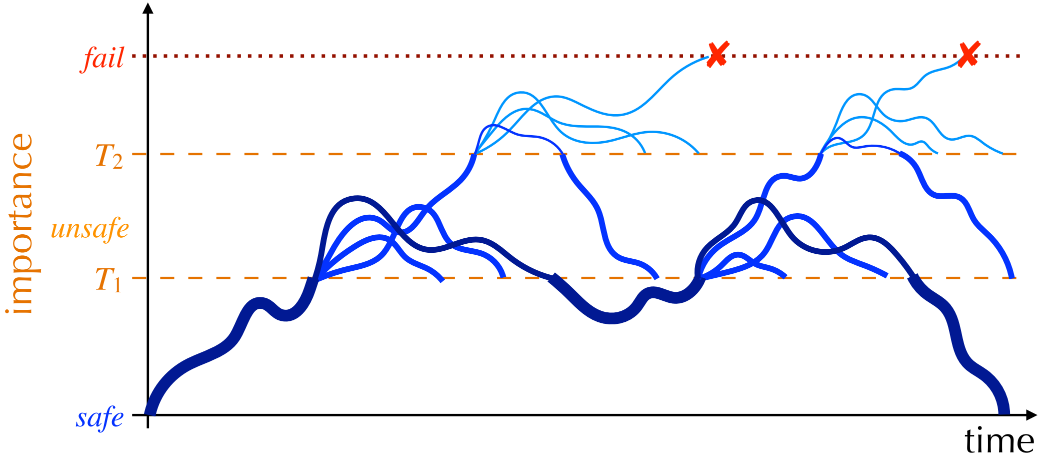
Rare event simulation through Importance Splitting



rare event

Ideally indicates the "proximity" to the rare event

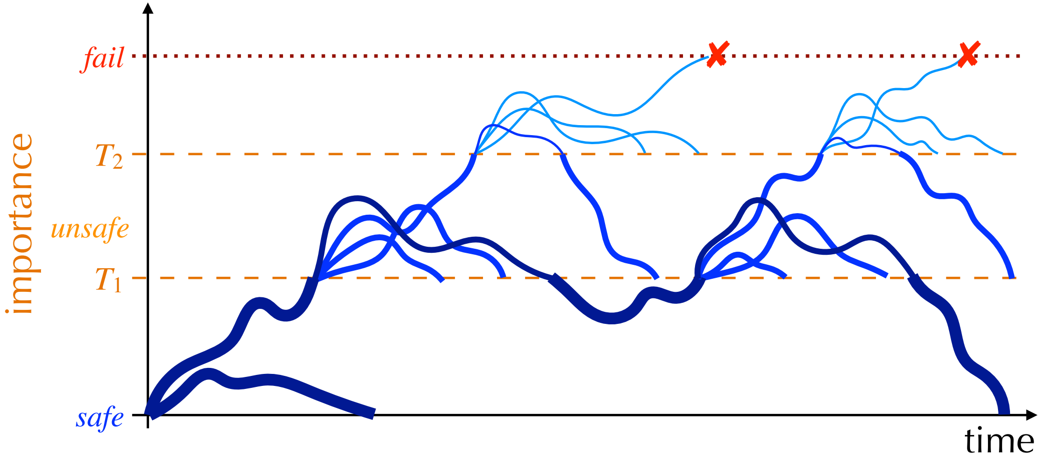
Rare event simulation through Importance Splitting



rare event

Ideally indicates the “proximity” to the rare event

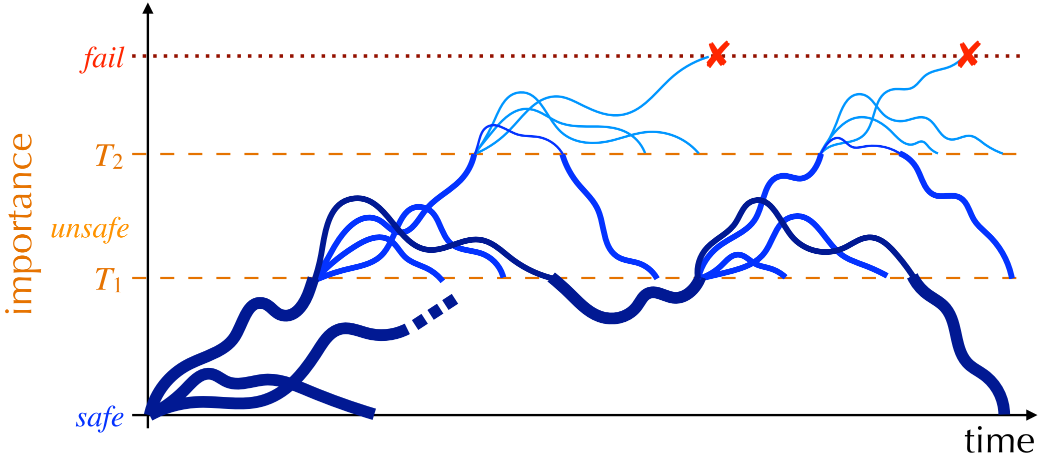
Rare event simulation through Importance Splitting



rare event

Ideally indicates the “proximity” to the rare event

Rare event simulation through Importance Splitting

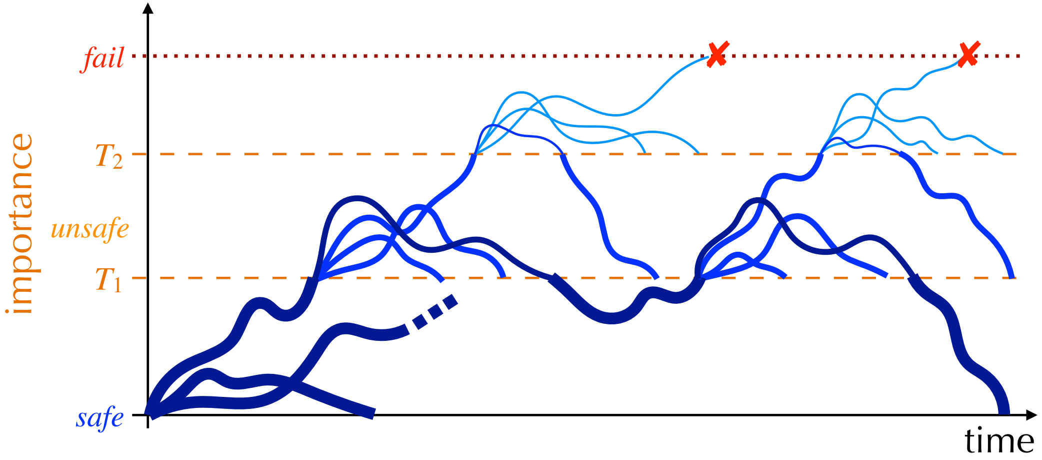


rare event

Ideally indicates the "proximity" to the rare event

Rare event simulation through Importance Splitting

Prob (*unsafe* U *fail*) ≈ $\hat{p}$



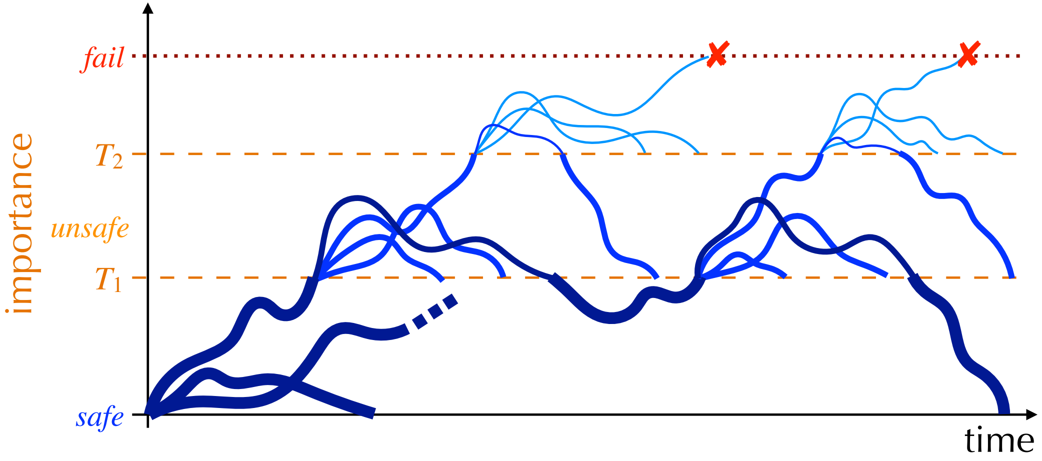
rare event

Ideally indicates the “proximity” to the rare event

Rare event simulation through Importance Splitting

$$\text{Prob}\,(\,\textit{unsafe}\ \mathsf{U}\ \textit{fail}\,) \approx \hat{p} = \frac{\#\text{✗}}{\#\text{total}}$$

rare event

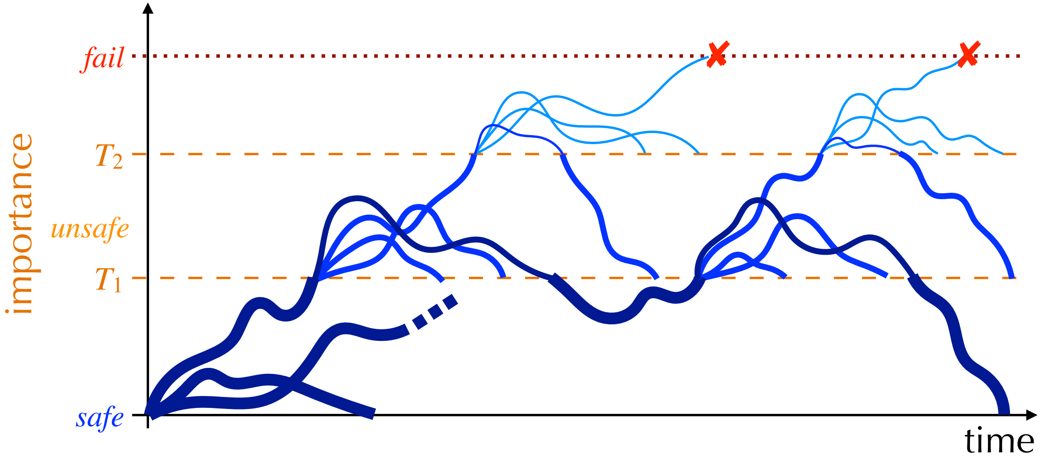


Ideally indicates the "proximity" to the rare event

Rare event simulation through Importance Splitting

$$\text{Prob}\,(\,\textit{unsafe}\ \mathsf{U}\ \textit{fail}\,) \approx \hat{p} = \frac{\#\text{✗}}{\#\text{total}} = \frac{\#\text{✗}}{\quad\quad\quad}$$

rare event

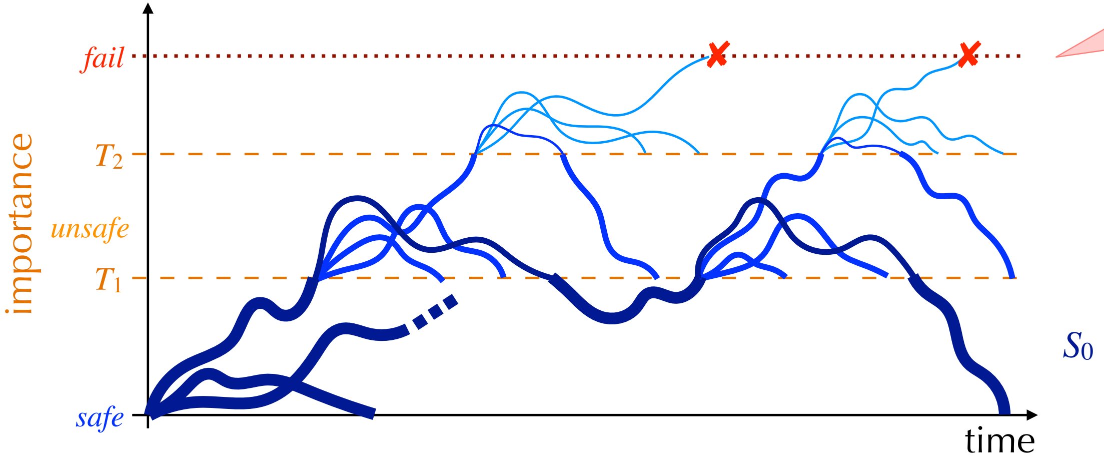


Ideally indicates the “proximity” to the rare event

Rare event simulation through Importance Splitting

$$\text{Prob}\,(\,\textit{unsafe}\ \mathsf{U}\ \textit{fail}\,) \approx \hat{p} = \frac{\#✗}{\#\text{total}} = \frac{\#✗}{S_0}$$

rare event

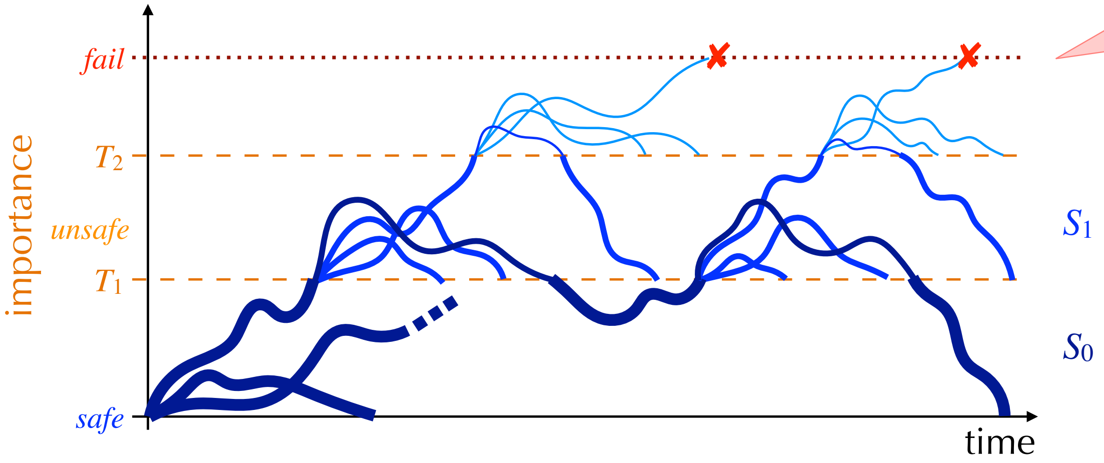


Ideally indicates the "proximity" to the rare event

Rare event simulation through Importance Splitting

$$\text{Prob}\,(\,\mathit{unsafe}\ \mathsf{U}\ \mathit{fail}\,) \approx \hat{p} = \frac{\#\text{✗}}{\#\text{total}} = \frac{\#\text{✗}}{S_0 * S_1}$$

rare event

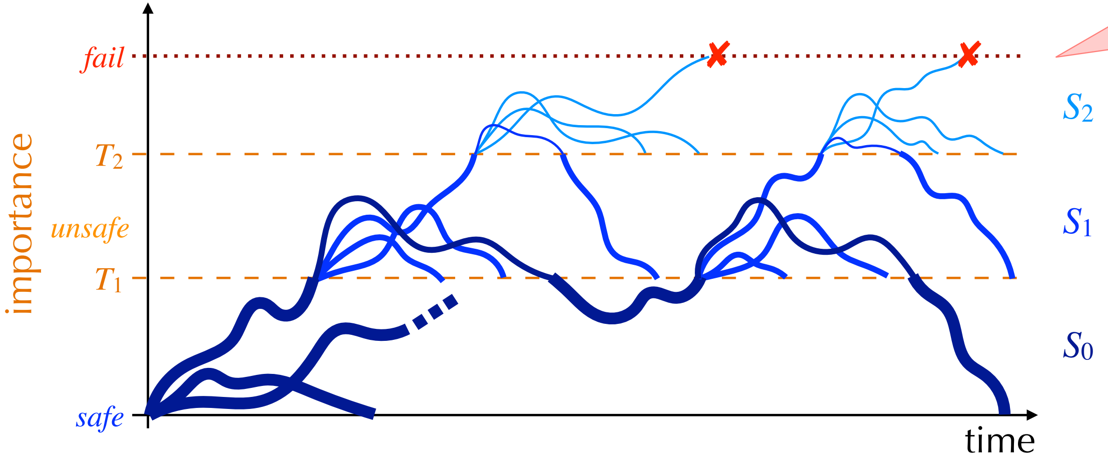


Ideally indicates the "proximity" to the rare event

Rare event simulation through Importance Splitting

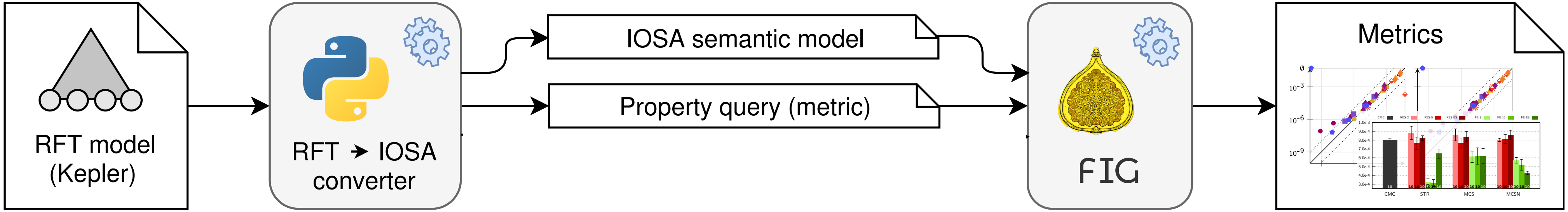
$$\mathrm{Prob}\,(\,\mathit{unsafe}\ \mathsf{U}\ \mathit{fail}\,) \approx \hat{p} = \frac{\#\text{✗}}{\#\text{total}} = \frac{\#\text{✗}}{S_0 * S_1 * S_2}$$

rare event

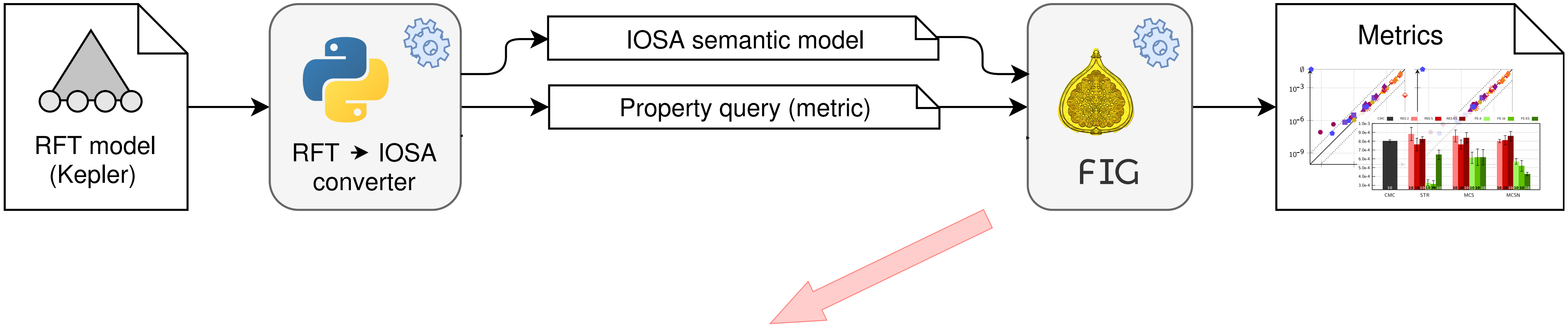


Ideally indicates the “proximity” to the rare event

Building the Tool Chain

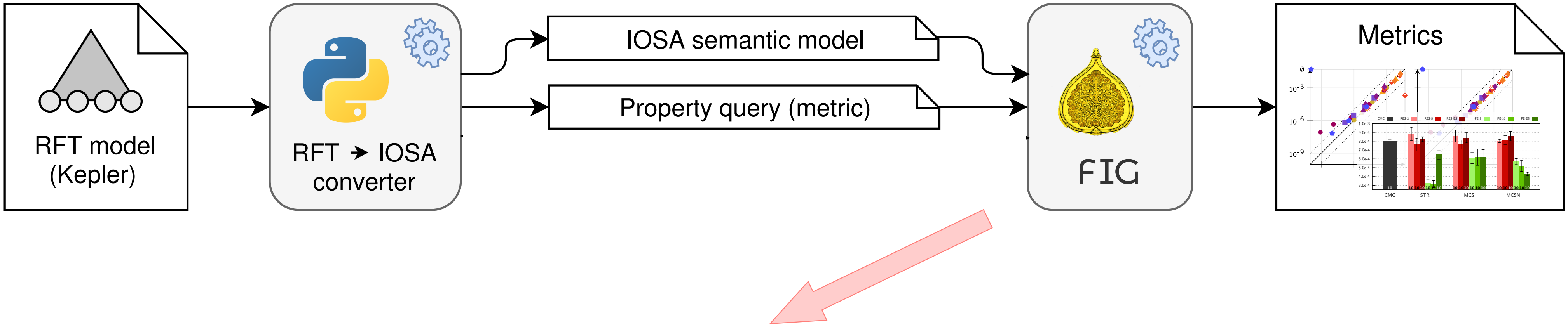


Building the Tool Chain



- ➡ importance function
- ➡ thresholds placing
- ➡ number of splittings

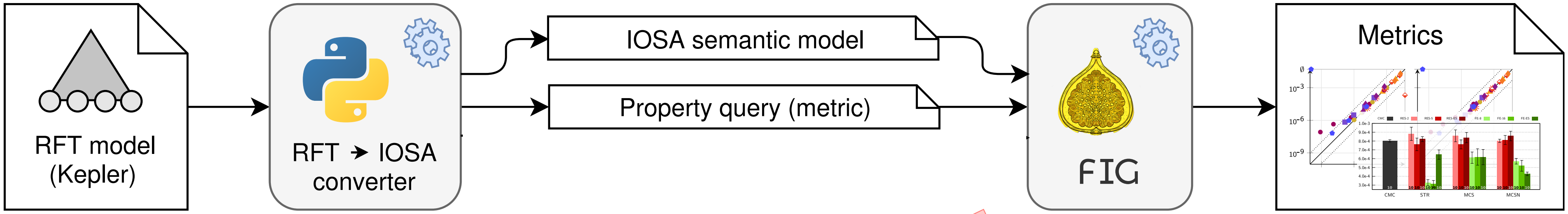
Building the Tool Chain



- importance function
- thresholds placing
- number of splittings

There are good strategies,

Building the Tool Chain



RFT model (Kepler)

RFT ➔ IOSA converter

IOSA semantic model

Property query (metric)

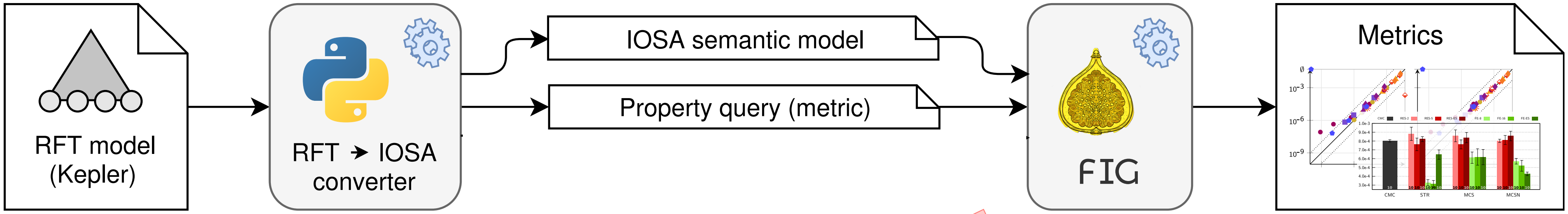
FIG

Metrics

- importance function
- thresholds placing
- number of splittings

There are good strategies, but they need

Building the Tool Chain



RFT model (Kepler)

RFT ➔ IOSA converter

IOSA semantic model

Property query (metric)

FIG

Metrics

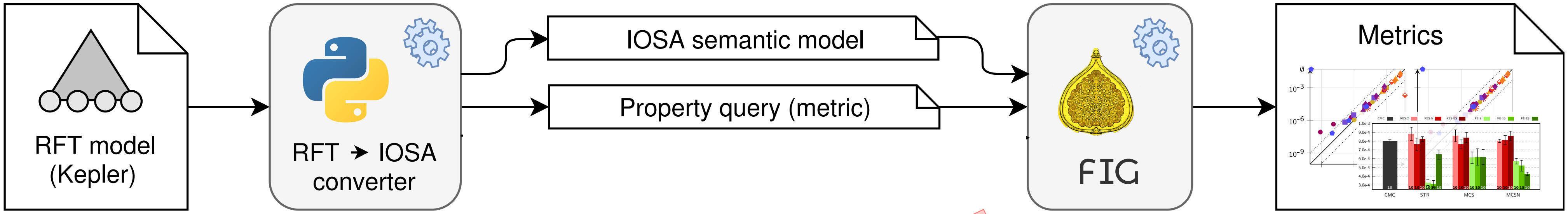
Provided in an ad hoc manner

- ➡ importance function
- ➡ thresholds placing
- ➡ number of splittings

There are good strategies, but they need

Building the Tool Chain

Fully Automatic



RFT model (Kepler)

RFT ➔ IOSA converter

IOSA semantic model

Property query (metric)

FIG

Metrics

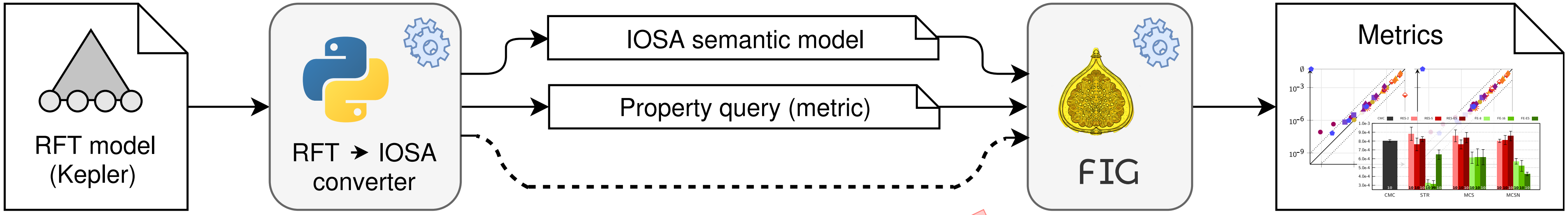
Provided in an ad hoc manner

- ➡ importance function
- ➡ thresholds placing
- ➡ number of splittings

There are good strategies, but they need

Building the Tool Chain

Fully Automatic

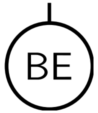


Provided in an
ad hoc manner

- ➡ importance function
- ➡ thresholds placing
- ➡ number of splittings

There are good
strategies, but
they need

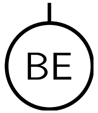
Deriving the importance function from RFT (the structural way)



$$\mathcal{I}_{\mathsf{BE}}(\vec{x}) = (\text{BE is failed}) \ ?\ 1 : 0$$

Deriving the importance function from RFT (the structural way)

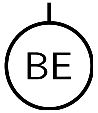
$\vec{x} \in \mathbb{N}^n$ is the state of the RFT with n nodes



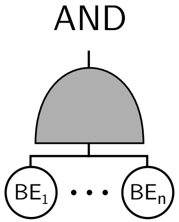
$$\mathcal{I}_{\mathsf{BE}}(\vec{x}) = (\text{BE is failed}) \; ? \; 1 : 0 = \vec{x}_{\mathsf{BE}}$$

Deriving the importance function from RFT (the structural way)

$\vec{x} \in \mathbb{N}^n$ is the state of the RFT with n nodes



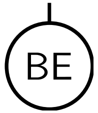
$\mathcal{I}_{\mathsf{BE}}(\vec{x}) = (\text{BE is failed}) \; ? \; 1 : 0 = \vec{x}_{\mathsf{BE}}$



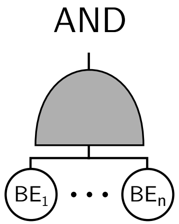
$\mathcal{I}_{\mathsf{AND}}(\vec{x}) = \sum_{w \in chil(\mathsf{AND})} \mathcal{I}_w(\vec{x})$

Deriving the importance function from RFT (the structural way)

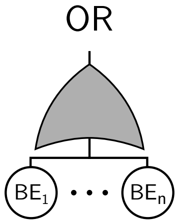
$\vec{x} \in \mathbb{N}^n$ is the state of the RFT with n nodes



$$\mathcal{I}_{\mathsf{BE}}(\vec{x}) = (\mathsf{BE\ is\ failed})\ ?\ 1 : 0 = \vec{x}_{\mathsf{BE}}$$



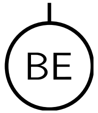
$$\mathcal{I}_{\mathsf{AND}}(\vec{x}) = \sum_{w \in chil(\mathsf{AND})} \mathcal{I}_w(\vec{x})$$



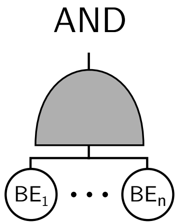
$$\mathcal{I}_{\mathsf{OR}}(\vec{x}) = \max_{w \in chil(\mathsf{OR})} \mathcal{I}_w(\vec{x})$$

Deriving the importance function from RFT (the structural way)

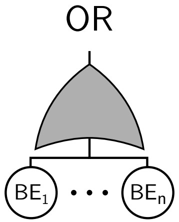
$\vec{x} \in \mathbb{N}^n$ is the state of the RFT with n nodes



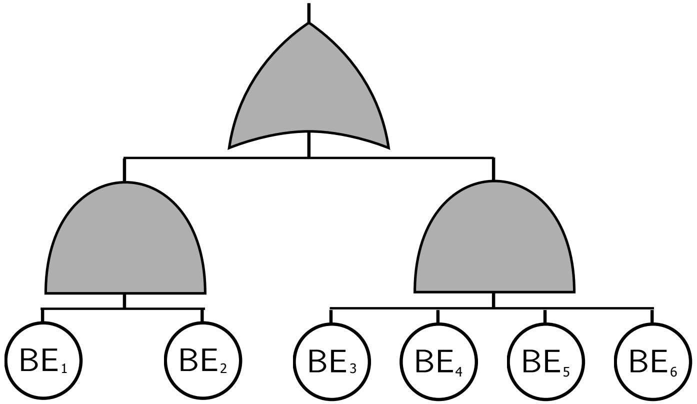
$$\mathcal{I}_{\text{BE}}(\vec{x}) = (\text{BE is failed}) \;?\; 1 : 0 = \vec{x}_{\text{BE}}$$



$$\mathcal{I}_{\text{AND}}(\vec{x}) = \sum_{w \in chil(\text{AND})} \mathcal{I}_w(\vec{x})$$



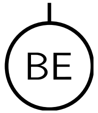
$$\mathcal{I}_{\text{OR}}(\vec{x}) = \max_{w \in chil(\text{OR})} \mathcal{I}_w(\vec{x})$$



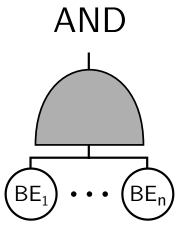
$$\mathcal{I}_{\text{OR}}(\vec{x}) =$$

Deriving the importance function from RFT (the structural way)

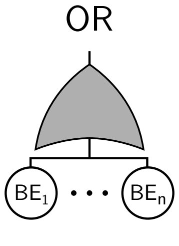
$\vec{x} \in \mathbb{N}^n$ is the state of the RFT with n nodes



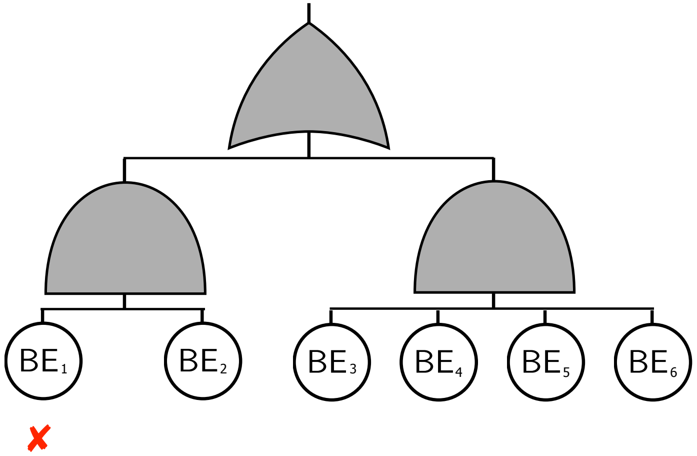
$$\mathcal{I}_{\mathsf{BE}}(\vec{x}) = (\text{BE is failed}) \; ? \; 1 : 0 = \vec{x}_{\mathsf{BE}}$$



$$\mathcal{I}_{\mathsf{AND}}(\vec{x}) = \sum_{w \in chil(\mathsf{AND})} \mathcal{I}_w(\vec{x})$$



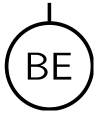
$$\mathcal{I}_{\mathsf{OR}}(\vec{x}) = \max_{w \in chil(\mathsf{OR})} \mathcal{I}_w(\vec{x})$$



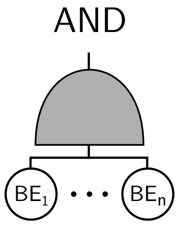
$$\mathcal{I}_{\mathsf{OR}}(\vec{x}) =$$

Deriving the importance function from RFT (the structural way)

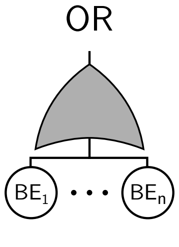
$\vec{x} \in \mathbb{N}^n$ is the state of the RFT with n nodes



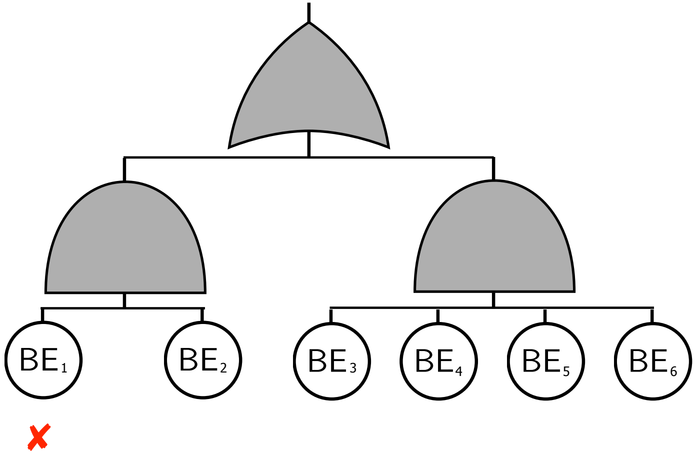
$$\mathcal{I}_{\text{BE}}(\vec{x}) = (\text{BE is failed}) \ ?\ 1 : 0 = \vec{x}_{\text{BE}}$$



$$\mathcal{I}_{\text{AND}}(\vec{x}) = \sum_{w \in chil(\text{AND})} \mathcal{I}_w(\vec{x})$$



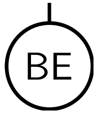
$$\mathcal{I}_{\text{OR}}(\vec{x}) = \max_{w \in chil(\text{OR})} \mathcal{I}_w(\vec{x})$$



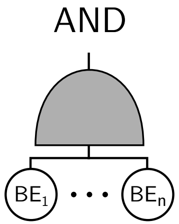
$$\mathcal{I}_{\text{OR}}(\vec{x}) = 1$$

Deriving the importance function from RFT (the structural way)

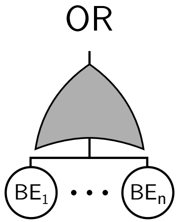
$\vec{x} \in \mathbb{N}^n$ is the state of the RFT with n nodes



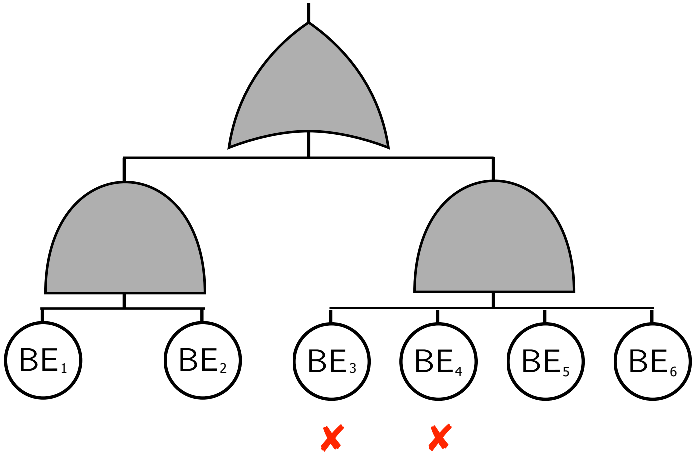
$\mathcal{I}_{\text{BE}}(\vec{x}) = (\text{BE is failed}) \;?\; 1 : 0 = \vec{x}_{\text{BE}}$



$\mathcal{I}_{\text{AND}}(\vec{x}) = \sum_{w \in chil(\text{AND})} \mathcal{I}_w(\vec{x})$



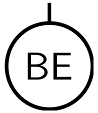
$\mathcal{I}_{\text{OR}}(\vec{x}) = \max_{w \in chil(\text{OR})} \mathcal{I}_w(\vec{x})$



$\mathcal{I}_{\text{OR}}(\vec{x}) =$

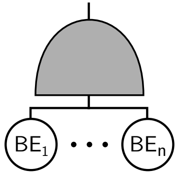
Deriving the importance function from RFT
(the structural way)

$\vec{x} \in \mathbb{N}^n$ is the state of the RFT with n nodes



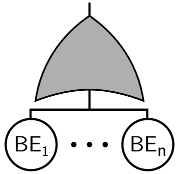
$$\mathcal{I}_{\mathsf{BE}}(\vec{x}) = (\text{BE is failed})\ ?\ 1 : 0 = \vec{x}_{\mathsf{BE}}$$

AND

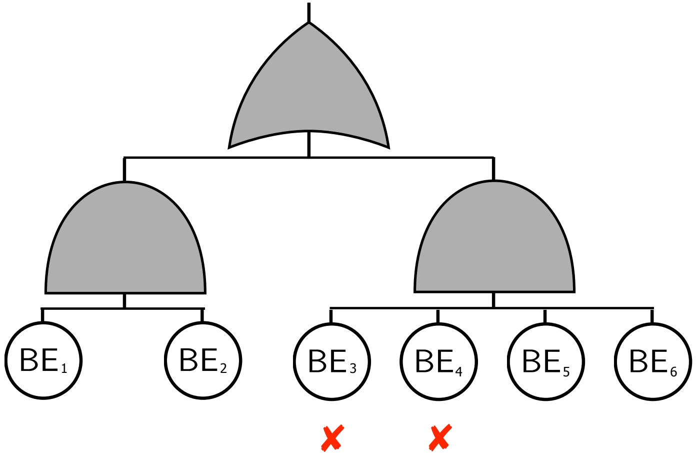


$$\mathcal{I}_{\mathsf{AND}}(\vec{x}) = \sum_{w\in chil(\mathsf{AND})} \mathcal{I}_w(\vec{x})$$

OR



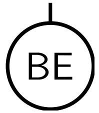
$$\mathcal{I}_{\mathsf{OR}}(\vec{x}) = \max_{w\in chil(\mathsf{OR})} \mathcal{I}_w(\vec{x})$$



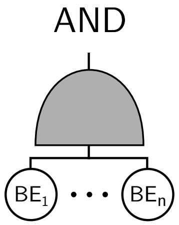
$$\mathcal{I}_{\mathsf{OR}}(\vec{x}) = 2$$

Deriving the importance function from RFT (the structural way)

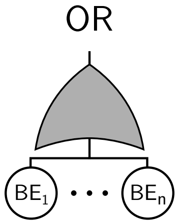
$\vec{x} \in \mathbb{N}^n$ is the state of the RFT with n nodes



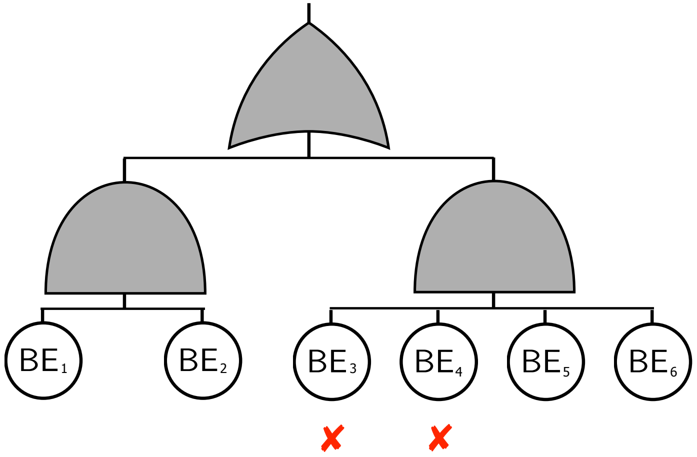
$$\mathcal{I}_{\mathsf{BE}}(\vec{x}) = (\text{BE is failed}) \; ? \; 1 : 0 = \vec{x}_{\mathsf{BE}}$$



$$\mathcal{I}_{\mathsf{AND}}(\vec{x}) = \sum_{w\in chil(\mathsf{AND})} \mathcal{I}_w(\vec{x})$$



$$\mathcal{I}_{\mathsf{OR}}(\vec{x}) = \max_{w\in chil(\mathsf{OR})} \mathcal{I}_w(\vec{x})$$



$$\mathcal{I}_{\mathsf{OR}}(\vec{x}) = 2$$

Normalize

Deriving the importance function from RFT (the structural way)

t$[v]$	$\mathcal{I}_v(\vec{x})$
be, sbe	$\vec{x}_v$
and	$\mathrm{lcm}_v \cdot \sum_{w \in chil(v)} \frac{\mathcal{I}_w(\vec{x})}{\max^{\mathcal{I}}_w}$
or	$\mathrm{lcm}_v \cdot \max\limits_{w \in chil(v)} \left\{ \frac{\mathcal{I}_w(\vec{x})}{\max^{\mathcal{I}}_w} \right\}$
vot$_k$	$\mathrm{lcm}_v \cdot \max\limits_{W \subseteq chil(v), \lvert W \rvert = k} \left\{ \sum_{w \in W} \frac{\mathcal{I}_w(\vec{x})}{\max^{\mathcal{I}}_w} \right\}$
sg	$\mathrm{lcm}_v \cdot \max \left(\sum_{w \in chil(v)} \frac{\mathcal{I}_w(\vec{x})}{\max^{\mathcal{I}}_w} \, , \, \vec{x}_v \cdot m \right)$
pand	$\mathrm{lcm}_v \cdot \max \left(\frac{\mathcal{I}_l(\vec{x})}{\max^{\mathcal{I}}_l} + ord \, \frac{\mathcal{I}_r(\vec{x})}{\max^{\mathcal{I}}_r} \, , \, \vec{x}_v \cdot 2 \right)$

where

$$\max^{\mathcal{I}}_v = \max_{\vec{x} \in \mathcal{S}} \mathcal{I}_v(\vec{x})$$

$$\mathrm{lcm}_v = \mathrm{lcm} \left\{ \max^{\mathcal{I}}_w \,\middle|\, w \in chil(v) \right\}$$

$$ord = \begin{cases} 1 & \text{if } \vec{x}_v \in \{1, 4\} \\ -1 & \text{otherwise} \end{cases}$$

Deriving the importance function from RFT (the structural way)

t$[v]$	$\mathcal{I}_v(\vec{x})$
be, sbe	$\vec{x}_v$
and	$\mathrm{lcm}_v \cdot \sum_{w\in chil(v)} \frac{\mathcal{I}_w(\vec{x})}{\max^{\mathcal{I}}_w}$
or	$\mathrm{lcm}_v \cdot \max_{w\in chil(v)} \left\{ \frac{\mathcal{I}_w(\vec{x})}{\max^{\mathcal{I}}_w} \right\}$
vot$_k$	$\mathrm{lcm}_v \cdot \max_{W\subseteq chil(v), \lvert W\rvert=k} \left\{ \sum_{w\in W} \frac{\mathcal{I}_w(\vec{x})}{\max^{\mathcal{I}}_w} \right\}$
sg	$\mathrm{lcm}_v \cdot \max \left(\sum_{w\in chil(v)} \frac{\mathcal{I}_w(\vec{x})}{\max^{\mathcal{I}}_w} \, , \, \vec{x}_v \cdot m \right)$
pand	$\mathrm{lcm}_v \cdot \max \left(\frac{\mathcal{I}_l(\vec{x})}{\max^{\mathcal{I}}_l} + ord \, \frac{\mathcal{I}_r(\vec{x})}{\max^{\mathcal{I}}_r} \, , \, \vec{x}_v \cdot 2 \right)$

where

$$\max^{\mathcal{I}}_v = \max_{\vec{x}\in\mathcal{S}} \mathcal{I}_v(\vec{x})$$

$$\mathrm{lcm}_v = \mathrm{lcm}\left\{ \max^{\mathcal{I}}_w \,\middle|\, w \in chil(v) \right\}$$

$$(\qquad 4\}$$

Check for updates

Rare event simulation for non-Markovian repairable fault trees ☆

Carlos E. Budde[1], Marco Biagi[2], Raúl E. Monti[1], Pedro R. D'Argenio[3,4,5], and Mariëlle Stoelinga[1,6]

[1] Formal Methods and Tools, University of Twente, Enschede, the Netherlands
[2] Department of Information Engineering, University of Florence, Florence, Italy
[3] FAMAF, Universidad Nacional de Córdoba, Córdoba, Argentina
[4] CONICET, Córdoba, Argentina
[5] Department of Computer Science, Saarland University, Saarbrücken, Germany
[6] Department of Software Science, Radboud University, Nijmegen, the Netherlands

fault trees (DFT) are widely adopted in industry
critical equipment. Since many sys-
dependability is often

[TACAS 2020]

CONICET

UNC

Deriving the importance function from RFT (via minimal cut sets)

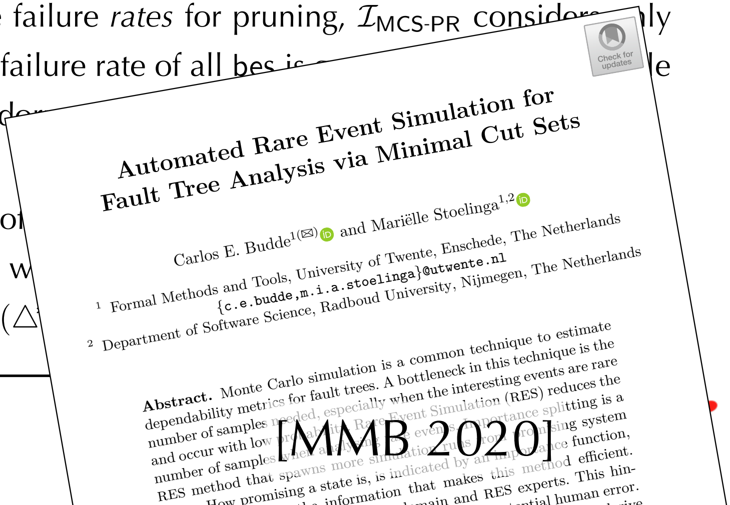
- Cut set: a set of BE that triggers a TLE (Top Level Event)
- It is minimal if removing any BE there is no TLE
- Originally defined for static fault trees
- We adapt them and extended to repairable fault trees but…
- If no PAND and Spare gates, all MCS can be collected
- If Spare gates but no PAND some MCS maybe lost for some configurations
- We did not include PAND

Deriving the importance function from RFT (via minimal cut sets)

Name	Expression	Description
$\mathcal{I}_{\textsf{MCS}}(\vec{x}) =$	$\max\limits_{\mathrm{MCS}\in\mathcal{M}(\triangle^*)} \left\{ \sum\limits_{v\in\mathrm{MCS}} \vec{x}_b \right\}$	For each MCS of the tree, $\mathcal{I}_{\textsf{MCS}}$ counts the number of bes that have failed in the current state $\vec{x}$. The importance $\mathcal{I}_{\textsf{MCS}}(\vec{x})$ of the current state of the tree is the maximum among these counts.
$\mathcal{I}_{\textsf{MCS-P}}(\vec{x}) =$	$\max\limits_{\mathrm{MCS}\in\mathcal{M}_{<N}(\triangle^*)} \left\{ \sum\limits_{v\in\mathrm{MCS}} \vec{x}_b \right\}$	$\mathcal{I}_{\textsf{MCS-P}}$ operates similarly to function $\mathcal{I}_{\textsf{MCS}}$ above, but here the maximum ranges over a *pruned* set of MCS, discarding cut sets with N or more bes.
$\mathcal{I}_{\textsf{MCS-PR}}(\vec{x}) =$	$\max\limits_{\mathrm{MCS}\in\mathcal{M}_{>\lambda}(\triangle^*)} \left\{ \sum\limits_{v\in\mathrm{MCS}} \vec{x}_b \right\}$	Similar to $\mathcal{I}_{\textsf{MCS-P}}$ but using the failure *rates* for pruning, $\mathcal{I}_{\textsf{MCS-PR}}$ considers only MCS where the product of the failure rate of all bes is greater than λ. Applicable only to FTs whose failure and dormancy distributions are Markovian.
$\mathcal{I}_{\textsf{MCSN}}(\vec{x}) =$	$\max\limits_{\mathrm{MCS}\in\mathcal{M}(\triangle^*)} \left\{ \mathrm{lcm} \cdot \sum\limits_{v\in\mathrm{MCS}} \frac{\vec{x}_b}{\lvert\mathrm{MCS}\rvert} \right\}$	$\mathcal{I}_{\textsf{MCSN}}$ is a normalised version of $\mathcal{I}_{\textsf{MCS}}$. The normalisation follows a similar procedure to the structured case, where lcm is the least common multiple of the cardinality of every MCS in $\mathcal{M}(\triangle^*)$.

Deriving the importance function from RFT (via minimal cut sets)

Name	Expression	Description
$\mathcal{I}_{\textsf{MCS}}(\vec{x}) =$	$\max\limits_{\mathrm{MCS}\in\mathcal{M}(\triangle^*)}\left\{\sum\limits_{v\in\mathrm{MCS}}\vec{x}_b\right\}$	For each MCS of the tree, $\mathcal{I}_{\textsf{MCS}}$ counts the number of bes that have failed in the current state $\vec{x}$. The importance $\mathcal{I}_{\textsf{MCS}}(\vec{x})$ of the current state of the tree is the maximum among these counts.
$\mathcal{I}_{\textsf{MCS-P}}(\vec{x}) =$	$\max\limits_{\mathrm{MCS}\in\mathcal{M}_{<N}(\triangle^*)}\left\{\sum\limits_{v\in\mathrm{MCS}}\vec{x}_b\right\}$	$\mathcal{I}_{\textsf{MCS-P}}$ operates similarly to function $\mathcal{I}_{\textsf{MCS}}$ above, but here the maximum ranges over a *pruned* set of MCS, discarding cut sets with N or more bes.
$\mathcal{I}_{\textsf{MCS-PR}}(\vec{x}) =$	$\max\limits_{\mathrm{MCS}\in\mathcal{M}_{>\lambda}(\triangle^*)}\left\{\sum\limits_{v\in\mathrm{MCS}}\vec{x}_b\right\}$	Similar to $\mathcal{I}_{\textsf{MCS-P}}$ but using the failure *rates* for pruning, $\mathcal{I}_{\textsf{MCS-PR}}$ considers ly MCS where the product of the failure rate of all bes is le only to FTs whose failure and de
$\mathcal{I}_{\textsf{MCSN}}(\vec{x}) =$	$\max\limits_{\mathrm{MCS}\in\mathcal{M}(\triangle^*)}\left\{\mathrm{lcm}\cdot\sum\limits_{v\in\mathrm{MCS}}\frac{\vec{x}_b}{\lvert\mathrm{MCS}\rvert}\right\}$	$\mathcal{I}_{\textsf{MCSN}}$ is a normalised version o cedure to the structured case, w cardinality of every MCS in $\mathcal{M}(\triangle$



Check for updates

Automated Rare Event Simulation for Fault Tree Analysis via Minimal Cut Sets

Carlos E. Budde[1](✉) and Mariëlle Stoelinga[1,2]

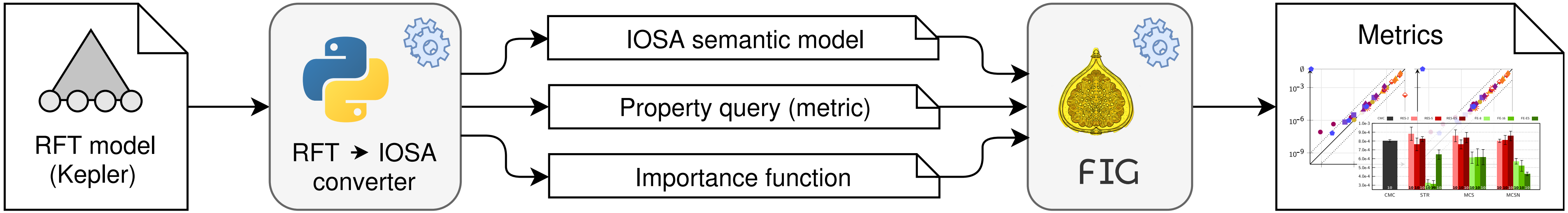
[1] Formal Methods and Tools, University of Twente, Enschede, The Netherlands
{c.e.budde,m.i.a.stoelinga}@utwente.nl

[2] Department of Software Science, Radboud University, Nijmegen, The Netherlands

Abstract. Monte Carlo simulation is a common technique to estimate dependability metrics for fault trees. A bottleneck in this technique is the number of samples needed, especially when the interesting events are rare and occur with low probability. Rare Event Simulation (RES) reduces the number of samples when analysing rare events. Importance splitting is a RES method that spawns more simulation runs from promising system states. How promising a state is, is indicated by an importance function, information that makes this method efficient. domain and RES experts. This hin- potential human error.

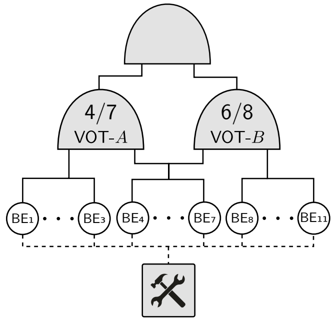
[MMB 2020]

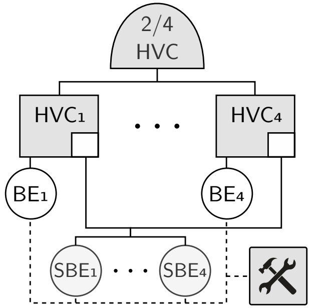
Building the Tool Chain

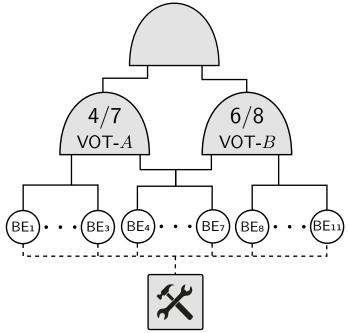


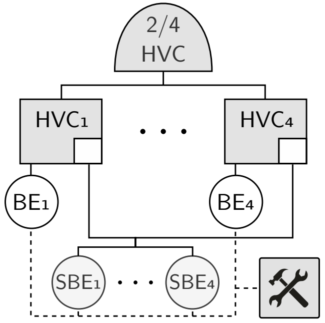
Fully Automatic!

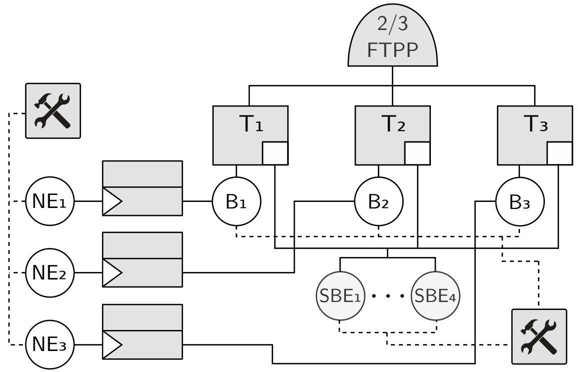
Experiments (Case Studies)

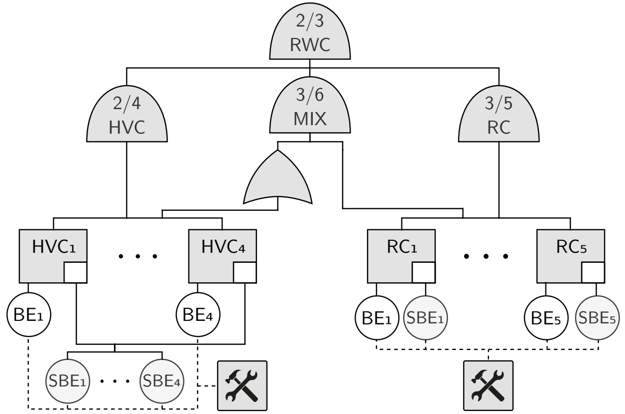


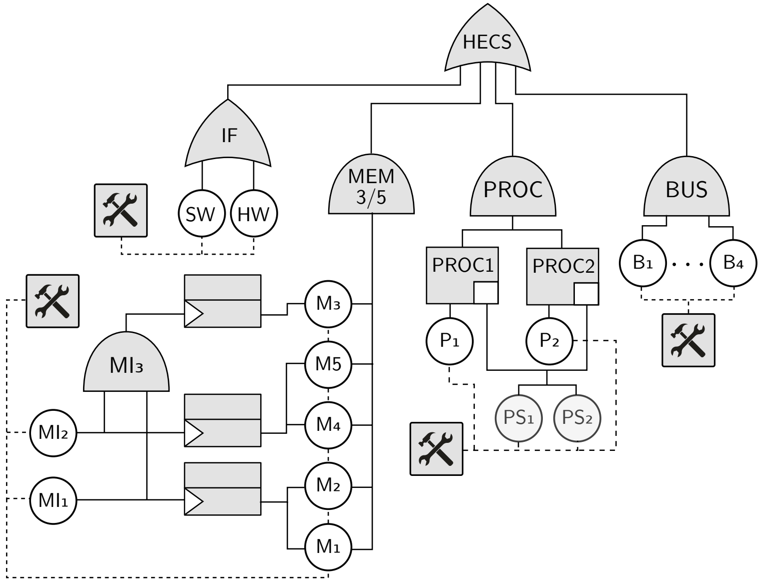












Experiments

(Case Studies)

Basic element	Fail time PDF	Repair PDF	Dormancy PDF
VOT:			
BE-A	$\text{lnor}(4.37, 0.33)$	$\text{uni}(0.4, 0.95)$	
BE-B	$\text{wei}(4.5, 0.0125)$	$\text{uni}(0.4, 0.95)$	
DSPARE:			
BE	$\exp(0.07)$	$\text{uni}(1.0, 2.0)$	
SBE	$\exp(0.07)$	$\text{uni}(1.0, 2.0)$	$\exp(0.035)$
HECS:			
SW	$\exp(4.5{\times}10^{-12})$	$\text{uni}(28.0, 56.0)$	
HW	$\exp(1.0{\times}10^{-10})$	$\text{uni}(28.0, 56.0)$	
MI_i	$\exp(5.0{\times}10^{-9})$	$\text{uni}(21.0, 28.0)$	
M_j	$\exp(6.0{\times}10^{-8})$	$\text{uni}(21.0, 28.0)$	
B_k	$\exp(8.7{\times}10^{-4})$	$\text{lnor}(4.45, 0.24)$	
P_a	$\exp(1.0{\times}10^{-3})$	$\text{lnor}(4.45, 0.24)$	
PS_b	$\exp(1.5{\times}10^{-3})$	$\text{lnor}(4.45, 0.24)$	$\text{dir}(\infty)$
FTPP:			
NE_i	$\text{lnor}(6.5, 0.5)$	$\text{nor}(150.0, 50.0)$	
B_j	$\exp(2.8{\times}10^{-2})$	$\text{nor}(15.0, 3.0)$	
SBE$_k$	$\exp(2.8{\times}10^{-2})$	$\text{nor}(15.0, 3.0)$	$\text{dir}(\infty)$
RC:			
BE$_i$	$\exp(0.04)$	$\text{nor}(2.0, 0.7)$	
SBE$_j$	$\exp(0.04)$	$\text{nor}(2.0, 0.7)$	$\exp(0.5)$
HVC:			
BE$_i$	$\text{ray}(1.999)$	$\text{uni}(0.15, 0.45)$	
SBE$_j$	$\text{ray}(1.999)$	$\text{uni}(0.15, 0.45)$	$\text{erl}(3.0, 0.25)$

Abbrev:	Distribution:
$\text{dir}(x)$	$\text{Dirac}(x)$
$\exp(\lambda)$	$\text{exponential}(\lambda)$
$\text{erl}(k, \lambda)$	$\text{Erlang}(k, \lambda)$
$\text{uni}(a, b)$	$\text{uniform}([a, b]_\mathbb{R})$
$\text{ray}(\sigma)$	$\text{Rayleigh}(\sigma)$
$\text{wei}(k, \lambda)$	$\text{Weibull}(k, \lambda)$
$\text{nor}(\mu, \sigma)$	$\text{normal}(\mu, \sigma)$
$\text{lnor}(\mu, \sigma)$	$\text{log-normal}(\mu, \sigma)$

Experiments

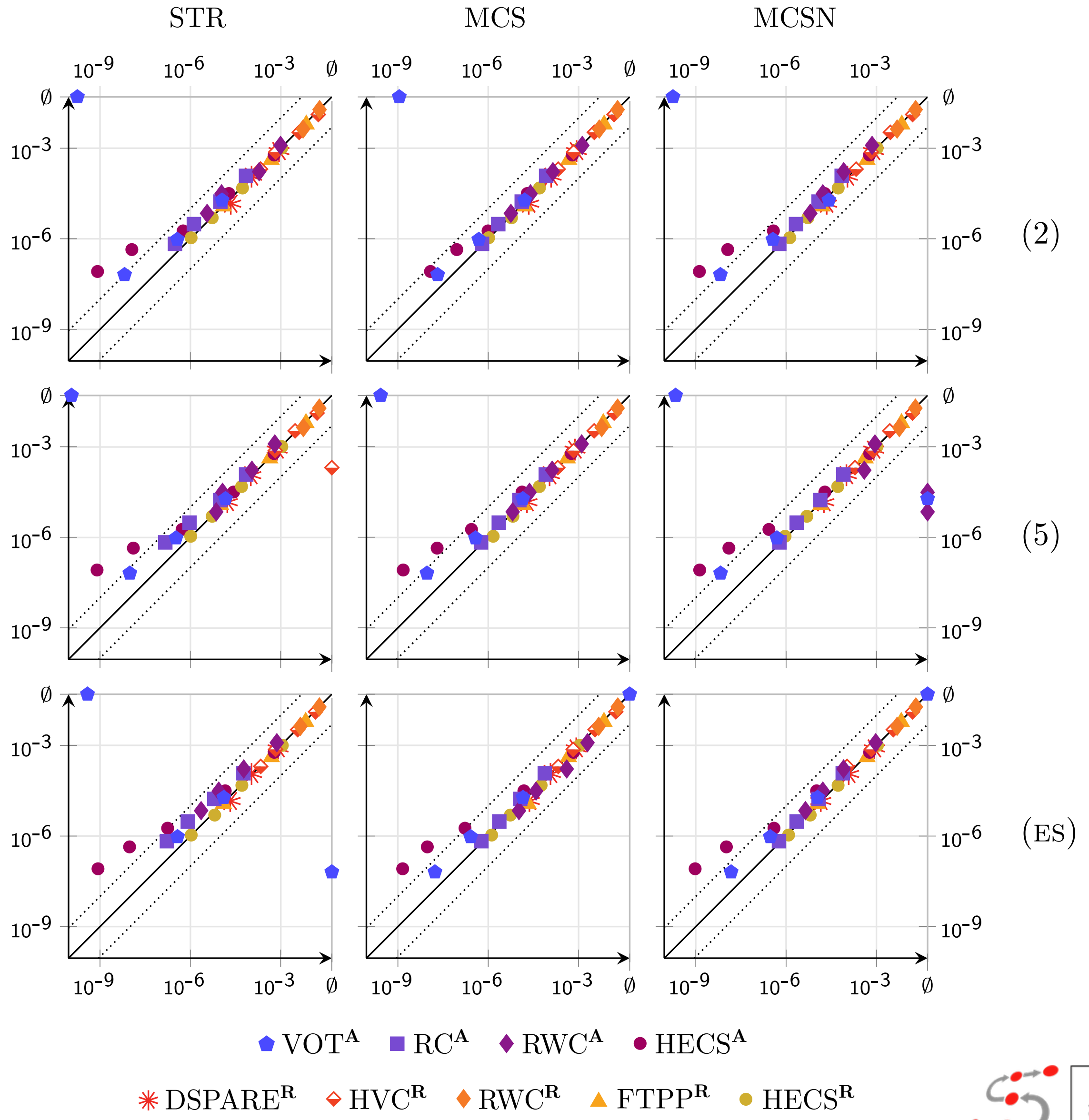
CMC

vs

RESTART

Availability

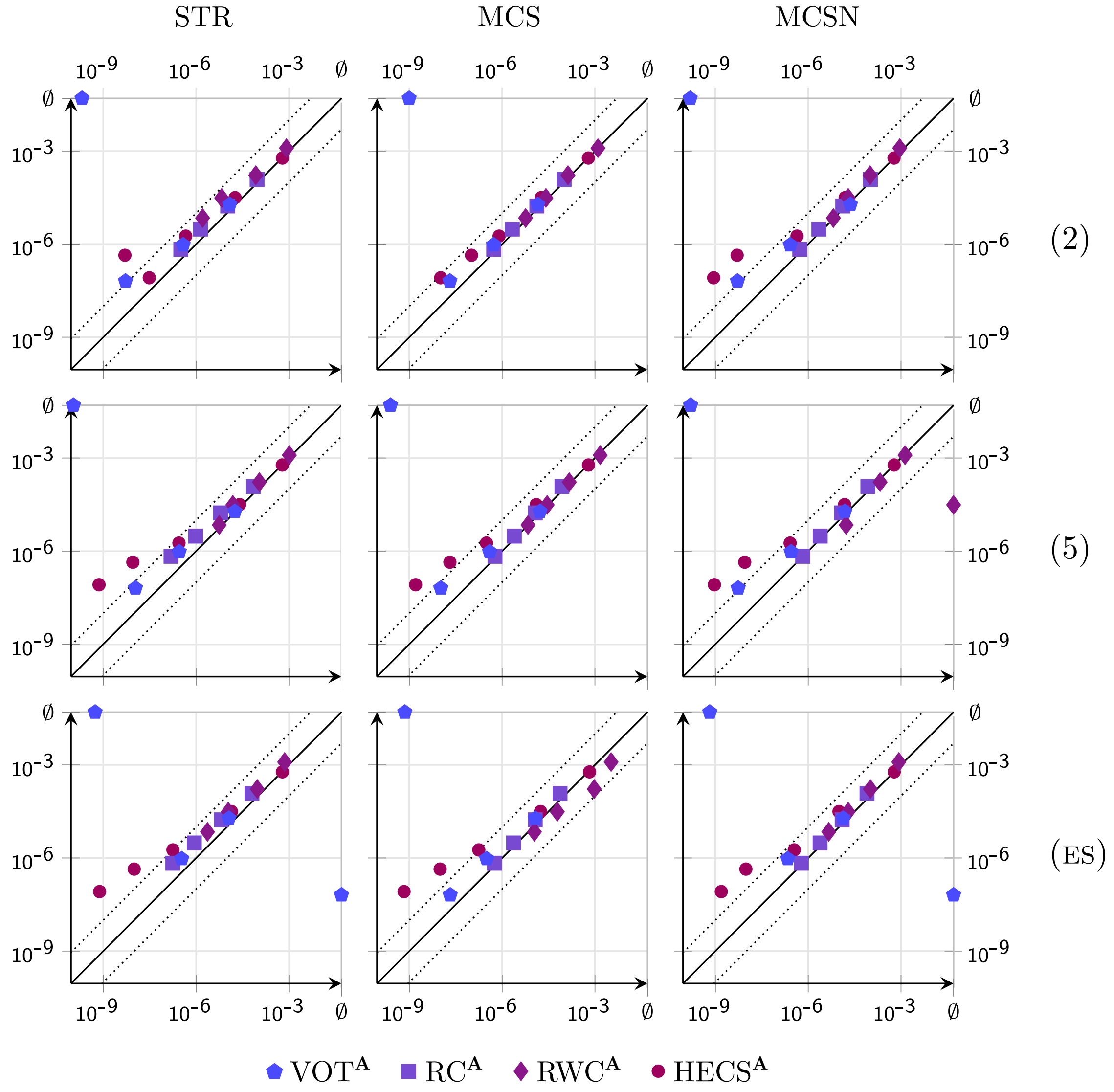
Reliability



Experiments

CMC
vs
RESTART-P2

Availability



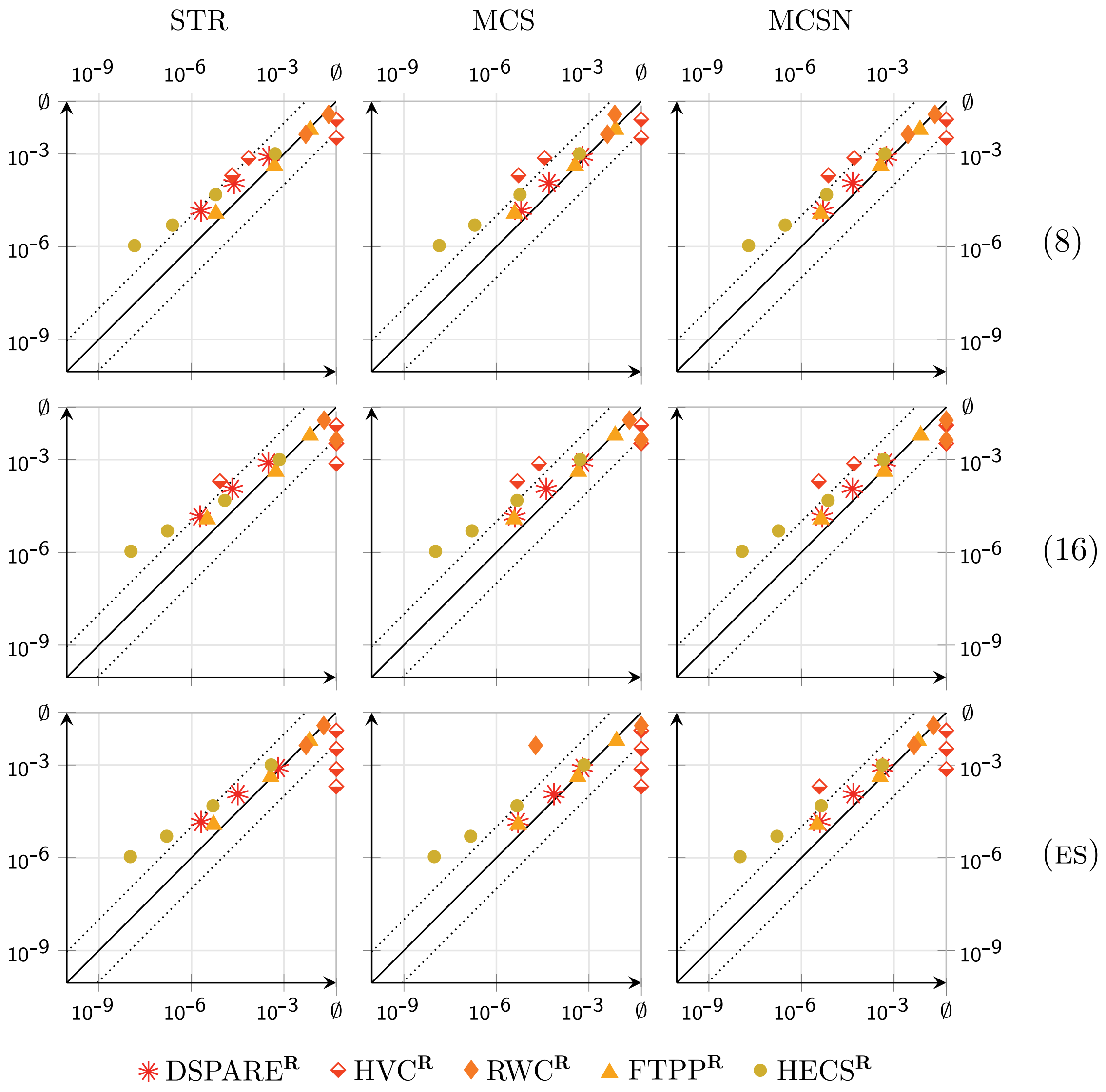
CONICET

UNC

Experiments

CMC
vs
Fixed Effort

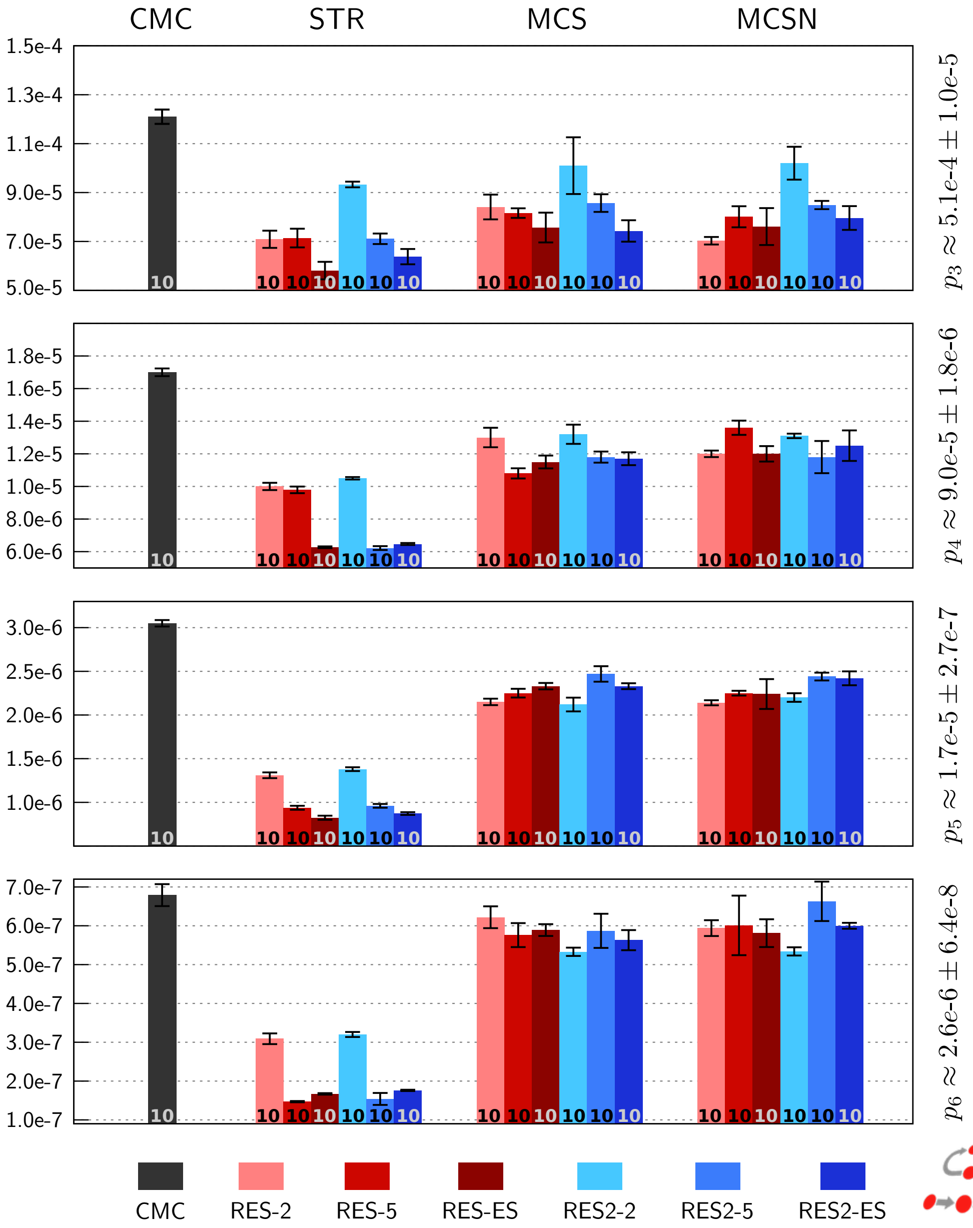
Reliability



Experiments

Availability

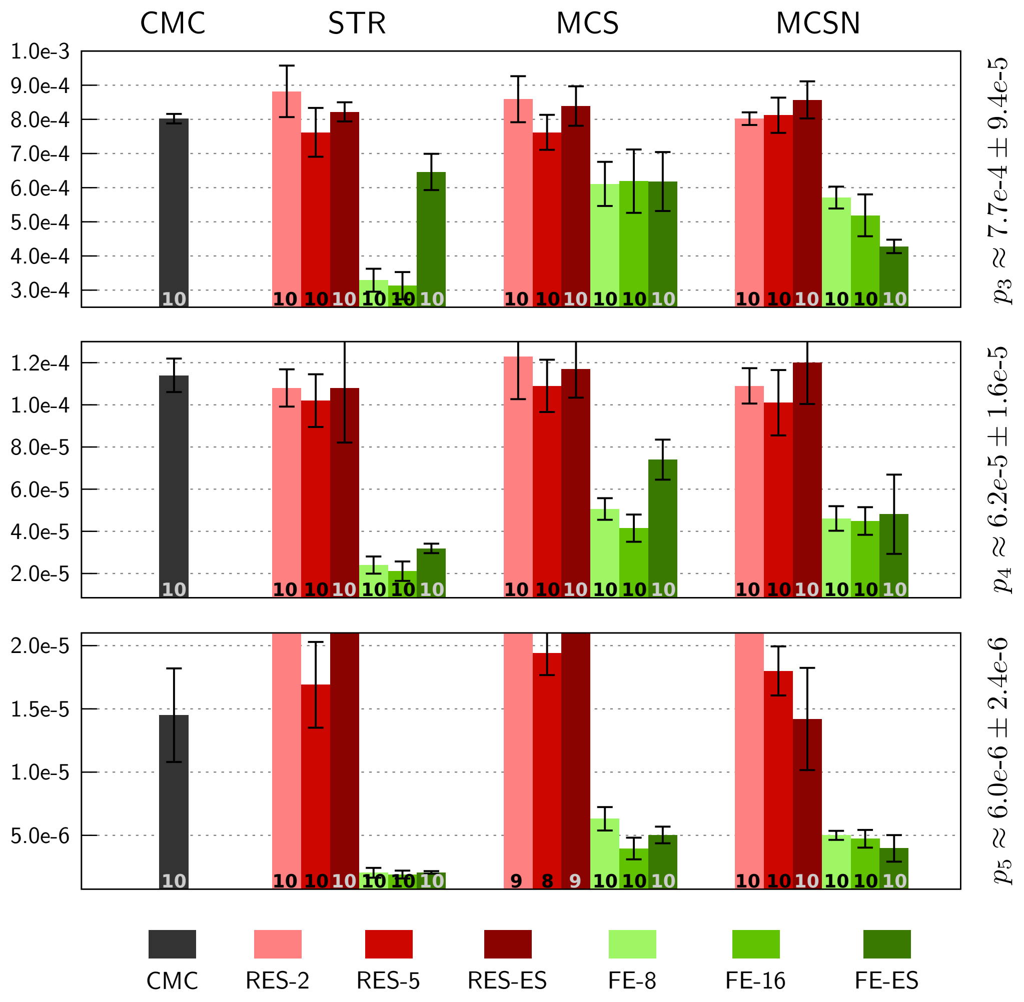
Case study: RC



Experiments

Reliability

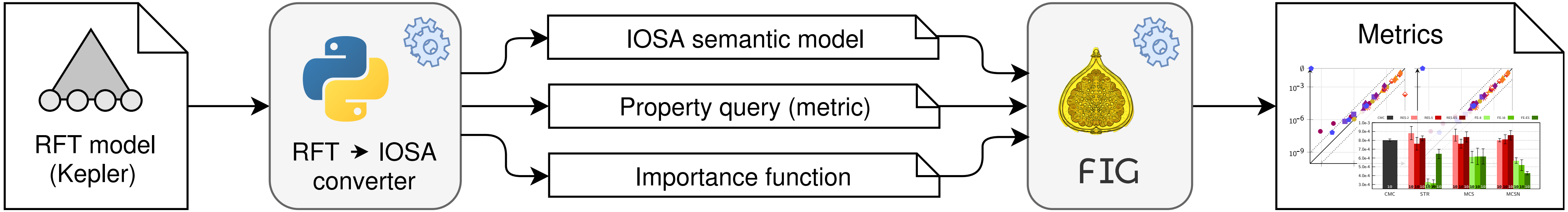
Case study:
DSPARE



CONICET

UNC

Final discussion



RFT model (Kepler)

RFT → IOSA converter

IOSA semantic model

Property query (metric)

Importance function

FIG

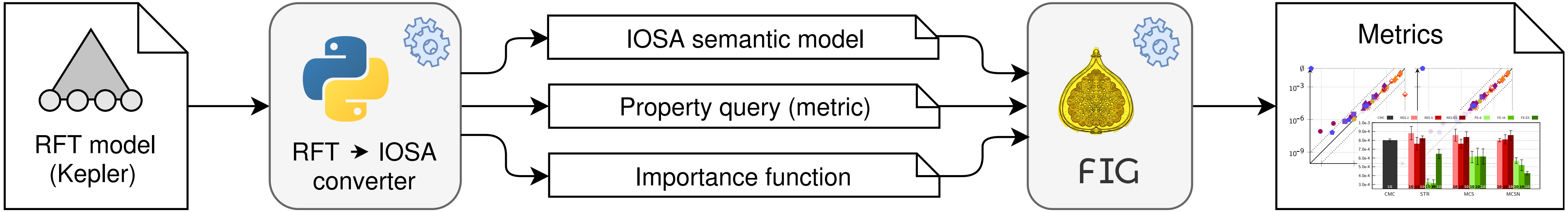
Metrics

CONICET

UNC

Final discussion

Fully Automatic



Dynamic Behaviour

Elements can be repaired

Large Systems

Arbitrary Distributions

Highly Reliable

Final discussion

- In general structural importance function showed the best performance
- MCS based important function occasionally performs worst than Monte Carlo
- Fixed effort showed better performance than RESTART (limited to reliability)
- … and work also well in combination with MCS based IF
- Still… not good enough (compare to importance sampling)
- Our importance functions are discrete
- Conjecture:

  if time and stochastics info is considered, continuous versions should work better

Final discussion

This work will appear in STTT

- In general structural importance function showed the best performance
- MCS based important function occasionally performs worst than Monte Carlo
- Fixed effort showed better performance than RESTART (limited to reliability)
- … and work also well in combination with MCS based IF
- Still… not good enough (compare to importance sampling)
- Our importance functions are discrete
- Conjecture:

  if time and stochastics info is considered, continuous versions should work better

CONICET

UNC

Analysis of Highly Reliable Repairable Fault Trees via Simulation

Pedro R. D’Argenio

Universidad Nacional de Córdoba – CONICET (AR)

Joint work with Carlos Budde, Raúl Monti, & Mariëlle Stoelinga

QEST 2022, Warsaw

Analysis of Highly Reliable Repairable Fault Trees via Simulation

Pedro R. D’Argenio

Universidad Nacional de Córdoba – CONICET (AR)

Join rlos B nti, & inga