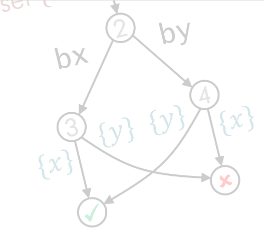
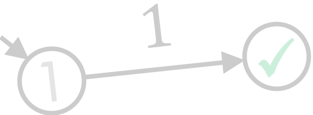
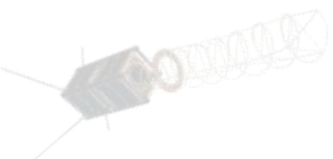
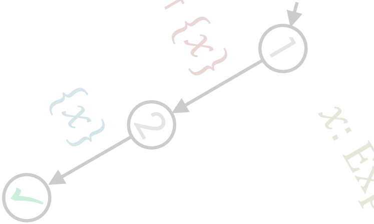
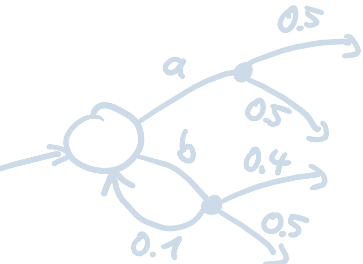


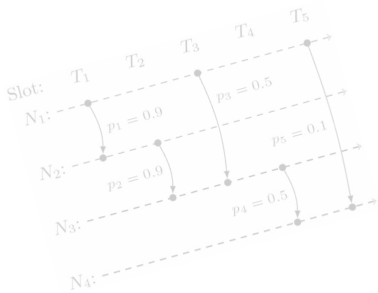


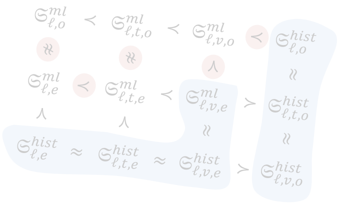


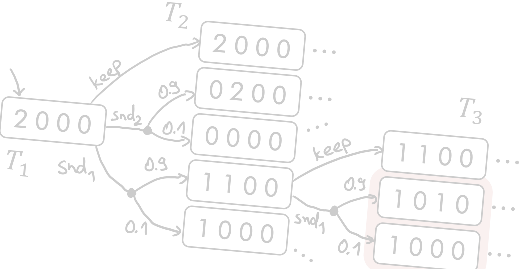


On Schedulers and Information in Distributed and Non-Markovian Probabilistic Systems









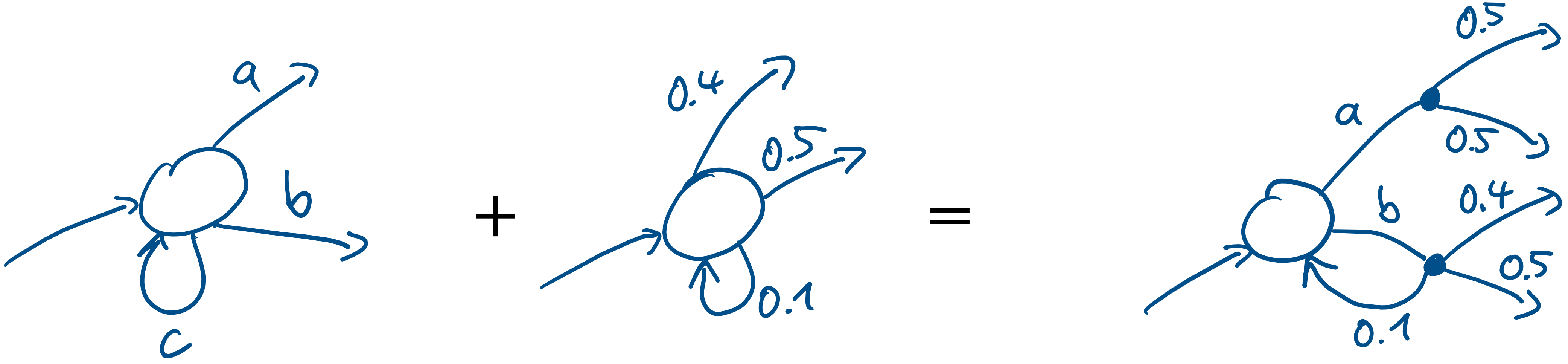
Arnd Hartmanns

University of Twente, Enschede, The Netherlands

based on joint work with Pedro R. D'Argenio, Juan A. Fraire, Marcus Gerhold, Holger Hermanns, Jan Krčál, Fernando Raverta, Sean Sedwards, and others

Probabilistic Systems

Nondeterminism and randomness:



LTS or similar + *DTMC* = *Markov decision process*

MDP

Properties of interest:

$\mathbb{S}$(power usage)

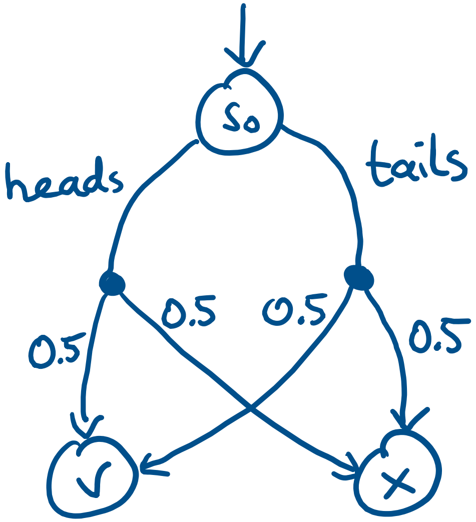
$\mathbb{P}(\diamond$ crash$)$

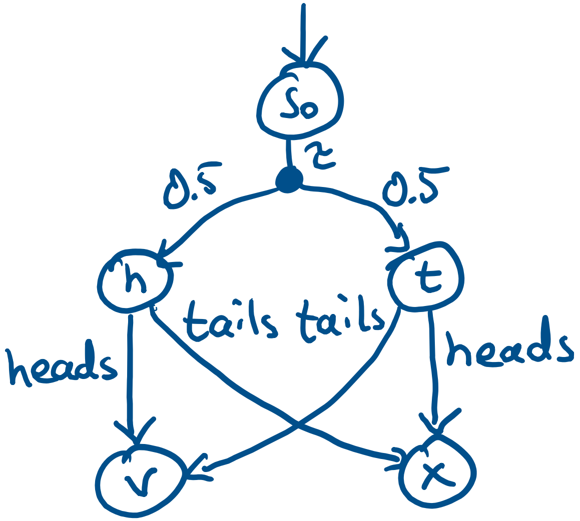
$\mathbb{E}$(time until message transmitted)

$\mathbb{P}(\diamond^{time\leq T}$ crash$)$

Schedulers

Coin throw: $\mathbb{P}(\diamond \checkmark) = ?$ *Scheduler:* resolves all nondeterminism





$\mathcal{S}: S \to A$ = memoryless
deterministic

$\mathcal{S}: (S \times A)^* \times S \to \mathrm{Dist}(A)$
= history-dependent
randomised

$\mathbb{P}(\diamond\, G)$: memoryless deterministic suffices

$\mathbb{P}\left(\diamond^{time \leq T}\, G\right)$: need remaining time to bound T

multi-objective: need randomised schedulers

find smallest
class sufficient
for property

1 schedulers and information in distributed probabilistic systems

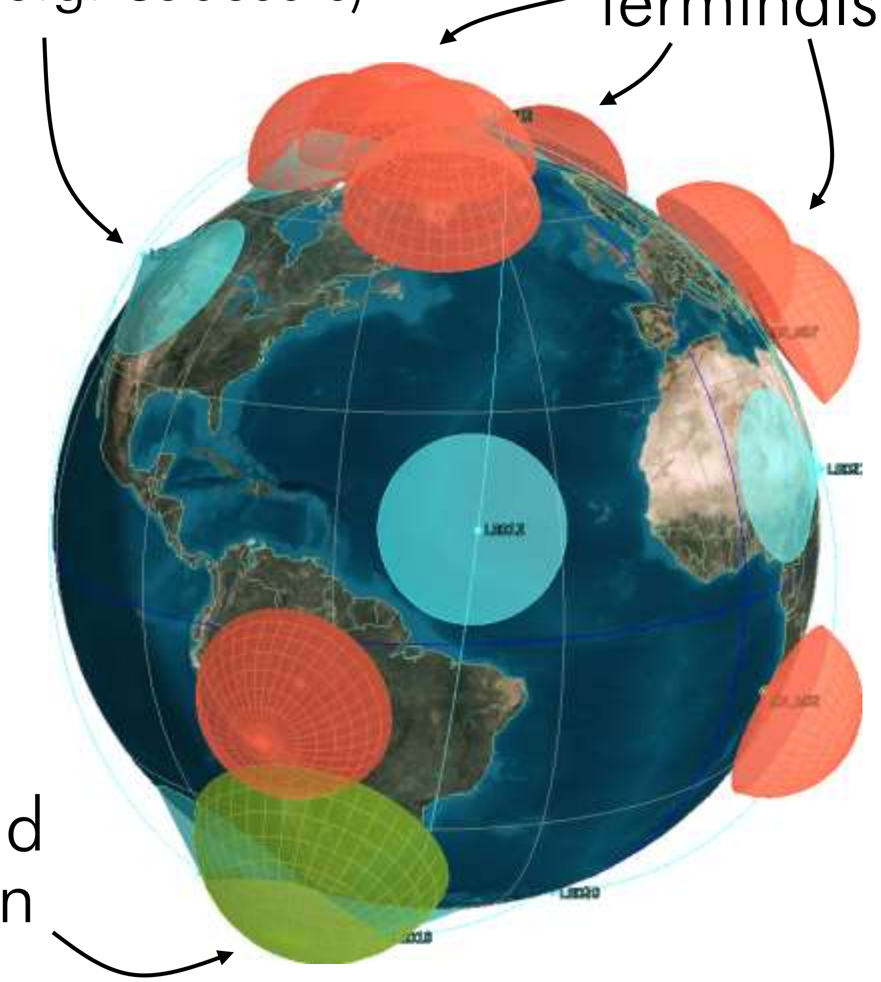
Uncertain Delay-Tolerant Networks

Store-and-carry-forward communication
→ data exchange in contact windows

Contacts known, resources limited,
uncertain transmissions:

allow k copies of message
throughout entire network

→ find route maximising end-to-end
successful delivery probability



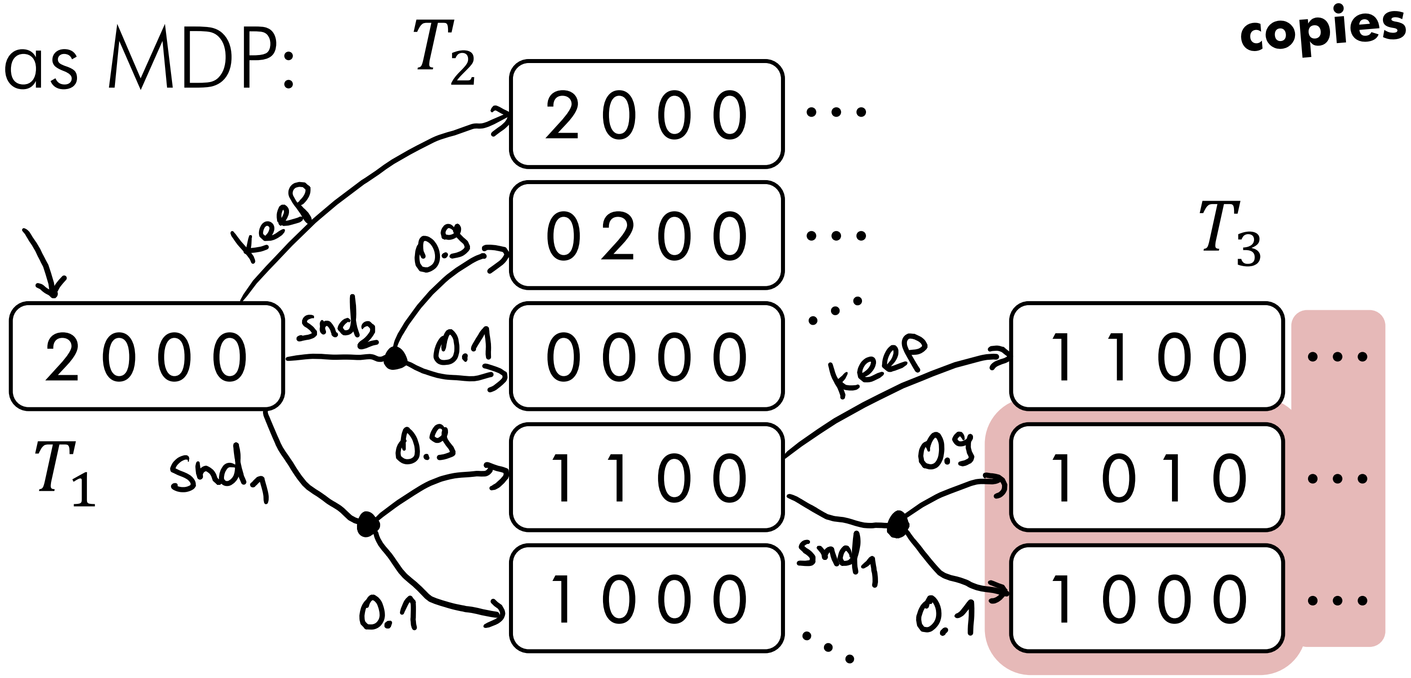
LEO satellites
(e.g. cubesats)

ground
terminals

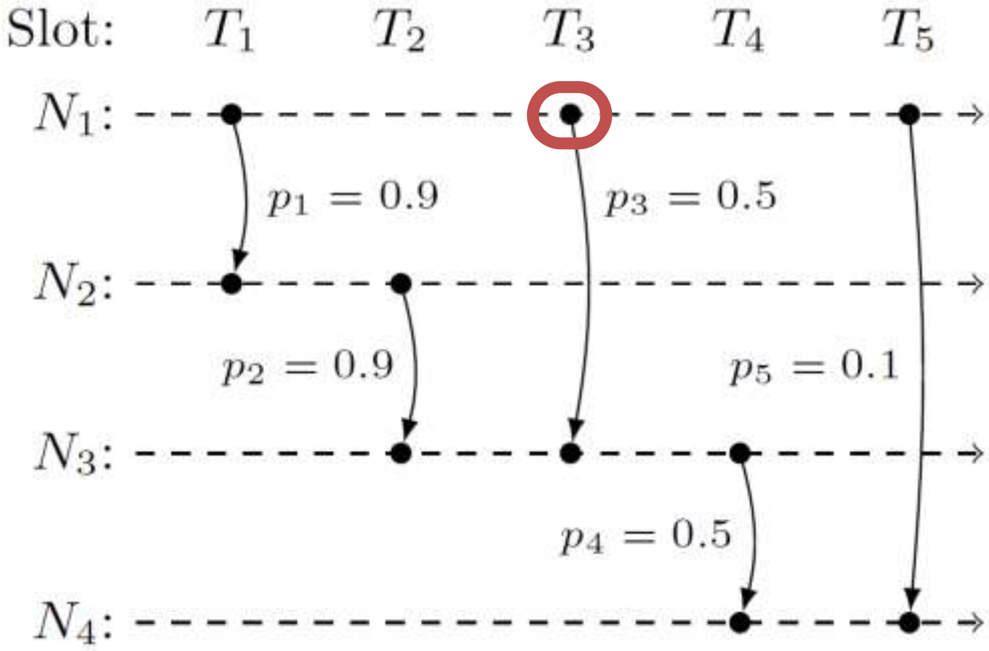
ground
station

Uncertain Contact Plans

as MDP:



max 2 copies



Best choice for N_1 in T_3 ($\mathcal{S}\colon S \to A$):

send remaining copy to N_3

only if N_3 did not get copy #1

global information

N_1 *cannot know this!*

Distributed Schedulers

Known problem in probabilistic model checking,$^{\text{PMC}}$
different from partial information (POMDP): multiple agents/views

Results from the 2000s: undecidable or NP-hard, no usable tools ☹

- 💡 L-RUCoP: local-information model checking algorithm specific to the uncertain DTN setting, approximates (Raverta et al., 2021)
- 💡 LSS: SMC with lightweight scheduler sampling among distributed schedulers

(D'Argenio, Fraire, H., NFM 2020)

Ad Hoc Networks 123 (2021) 102663

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

ELSEVIER

Routing in Delay-Tolerant Networks under uncertain contact plans

Fernando D. Raverta[a,*], Juan A. Fraire[a,b], Pablo G. Madoery[a], Ramiro A. Demasi[a], Jorge M. Finochietto[a], Pedro R. D'Argenio[a,b]

[a] CONICET-UNC, Argentina, Córdoba, 5000, Argentina
[b] Department of Computer Science, Saarland University, Saarbrücken, 66123, Germany

ARTICLE INFO

ABSTRACT

Sampling Distributed Schedulers for Resilient Space Communication

Pedro R. D'Argenio[1,2,3], Juan A. Fraire[1,2,3], and Arnd Hartmanns[4](✉)

[1] CONICET, Córdoba, Argentina
[2] Saarland University, Saarbrücken, Germany
[3] Universidad Nacional de Córdoba, Córdoba, Argentina
[4] Universit

Quantitative Model Checking Revisited: Neither Decidable Nor Approximable*

Time-Bounded Reachability in Distributed

Theoretical Computer Science 538 (2014) 84–102

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

ELSEVIER

Distributed probabilistic input/output automata: Expressiveness, (un)decidability and algorithms☆

Sergio Giro[a,*], Pedro R. D'Argenio[b], Luis María Ferrer Fioriti[c]

[a] Fakultät für Informatik, Technische Universität München, Germany
[b] Universidad Nacional de Córdoba, FaMAF – CONICET, Córdoba, Argentina
[c] Department of Computer Science, Saarbrücken, Germany

Distributed Schedulers

$$\text{Results from the 2000s: } \overbrace{\text{undecidable or NP-hard, no usable tools}}^{\text{Why give up? Simplify the problem!}}$$

- 💡 Good-for-distributed-scheduling models (no interference between components)
- 💡 LSS: SMC with lightweight scheduler sampling among distributed schedulers

(D'Argenio, Fraire, H., NFM 2020)

Sampling Distributed Schedulers for Resilient Space Communication

Pedro R. D'Argenio[1,2,3], Juan A. Fraire[1,2,3], and Arnd Hartmanns[4](✉)

[1] CONICET, Córdoba, Argentina
[2] Saarland University, Saarbrücken, Germany
[3] Universidad Nacional de Córdoba, Córdoba, Argentina

Quantitative Model Checking Revisited: Neither Decidable Nor Approximable*

Time-Bounded Reachability in Distributed

Theoretical Computer Science 538 (2014) 84–102

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Distributed probabilistic input/output automata: Expressiveness, (un)decidability and algorithms

Sergio Giro[a,*], Pedro R. D'Argenio[b], Luis María Ferrer Fioriti[c]

Distributed Schedulers

Why give up? Simplify the problem!

Results from the 2000s: undecidable or NP-hard, no usable tools

! engineers don’t want the history-dependent belief-estimating randomised schedulers anyway!

We need

simple

compact

understandable

implementable

explainable

schedulers!

Quantitative Model Checking Revisited: Neither Decidable Nor Approximable*

Time-Bounded Reachability in Distributed

Theoretical Computer Science 538 (2014) 84–102

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

ELSEVIER

Distributed probabilistic input/output automata: Expressiveness, (un)decidability and algorithms

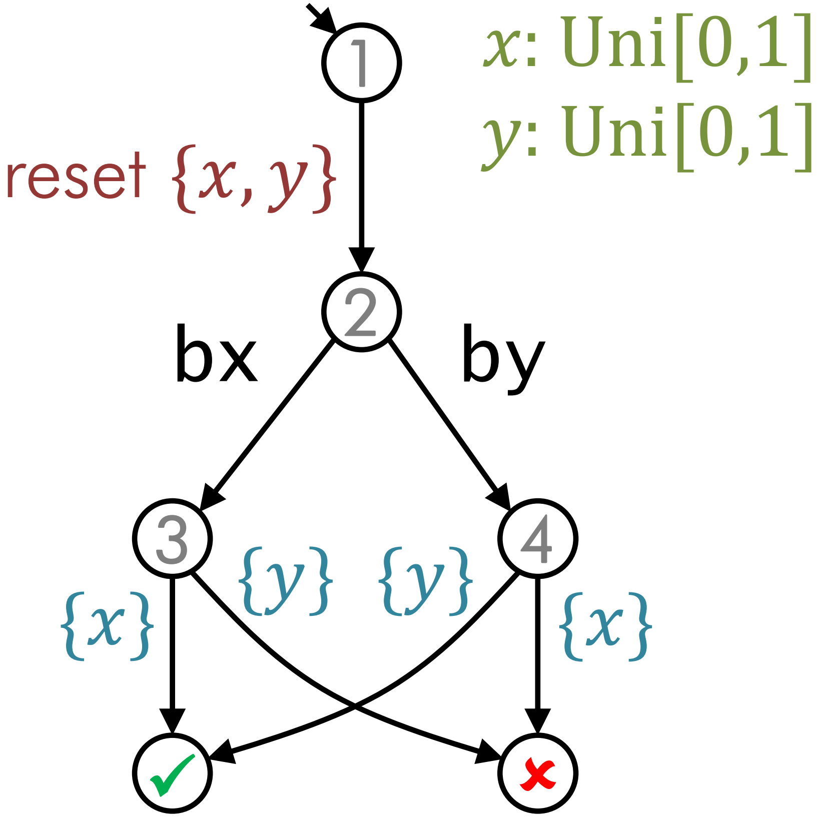
Sergio Giro, Pedro R. D'Argenio, Luis María Ferrer Fioriti

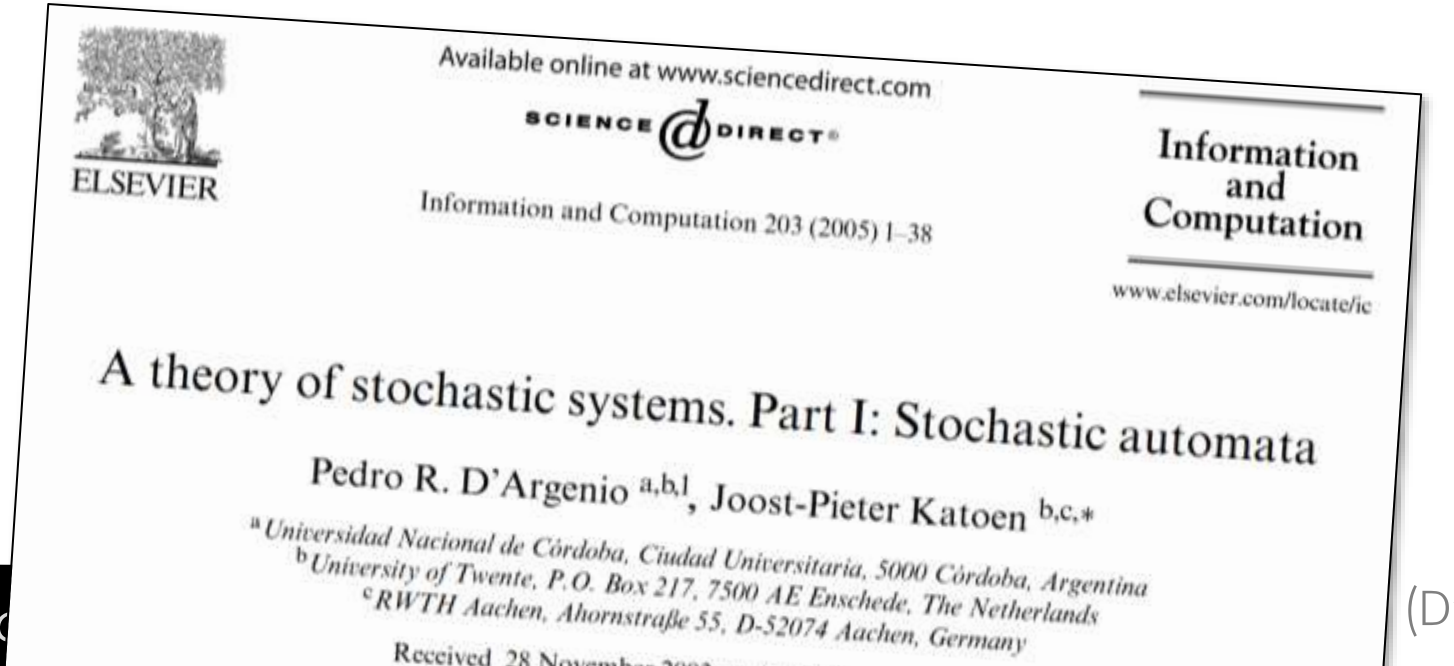
2 schedulers and information in non-Markovian probabilistic systems

Stochastic Automata

LTS extended with stochastic clocks that can be reset, and enable edge guards on expiry

symbolic model: need semantics





ELSEVIER

Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

Information and Computation 203 (2005) 1–38

Information and Computation

www.elsevier.com/locate/ic

A theory of stochastic systems. Part I: Stochastic automata

Pedro R. D'Argenio [a,b,1], Joost-Pieter Katoen [b,c,*]

[a] Universidad Nacional de Córdoba, Ciudad Universitaria, 5000 Córdoba, Argentina
[b] University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands
[c] RWTH Aachen, Ahornstraße 55, D-52074 Aachen, Germany

Received 28 Novembe

(D'Argenio & Katoen, 2005)

Stochastic Automata

Semantics: infinite-state "MDP"$^{\textbf{PTTS}}$
with **continuous** distributions
but **finite** nondeterminism

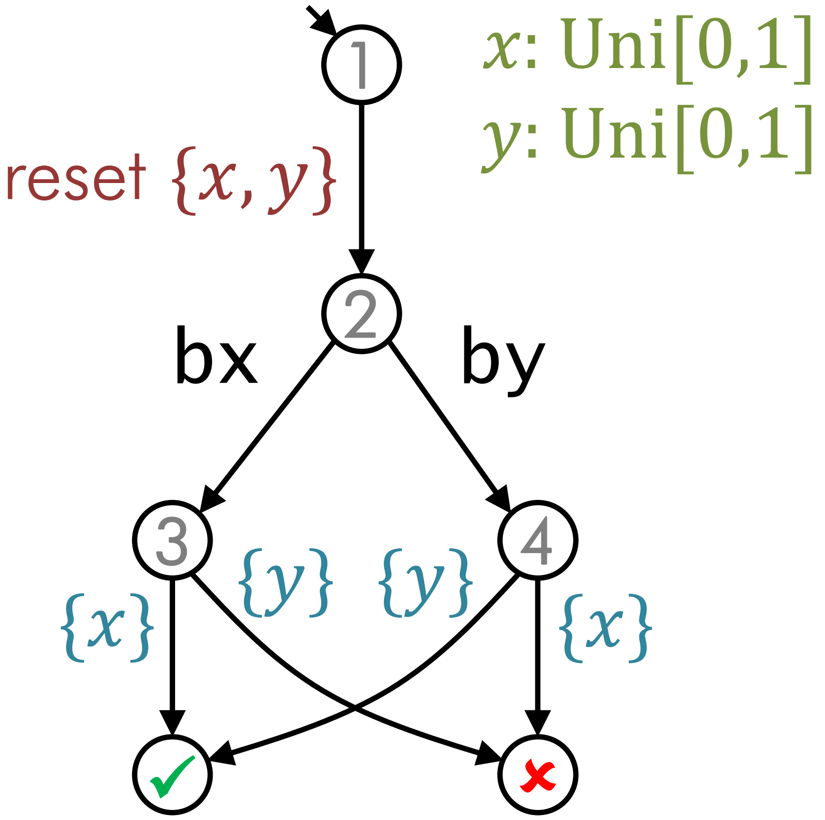
$$\to \left\langle 1, \overbrace{\langle 0,0\rangle}^{v(x),\, v(y)}, \overbrace{\langle 0,0\rangle}^{e(x),\, e(y)} \right\rangle$$

$$\to \left\langle 2, \langle 0,0\rangle, \langle 0.2,0.8\rangle \right\rangle \quad \text{Uni}[0,1]^2$$

$$\xrightarrow{\textbf{choice}} \left\langle 3, \langle 0,0\rangle, \langle 0.2,0.8\rangle \right\rangle$$

$$\to \left\langle 3, \langle 0.2,0.2\rangle, \langle 0.2,0.8\rangle \right\rangle$$

$$\to \left\langle ✓, \langle 0.2,0.2\rangle, \langle 0.2,0.8\rangle \right\rangle$$



→ *residual lifetimes* semantics
that turns STA into PTTS
admits prophetic schedulers

Fixing the Semantics

On every delay and jump,
resample unexpired clocks
conditioned on elapsed time

$$\to \left\langle 1, \overbrace{\langle 0,0\rangle}^{v(x),v(y)}, \overbrace{\langle 0,0\rangle}^{e(x),e(y)} \right\rangle$$

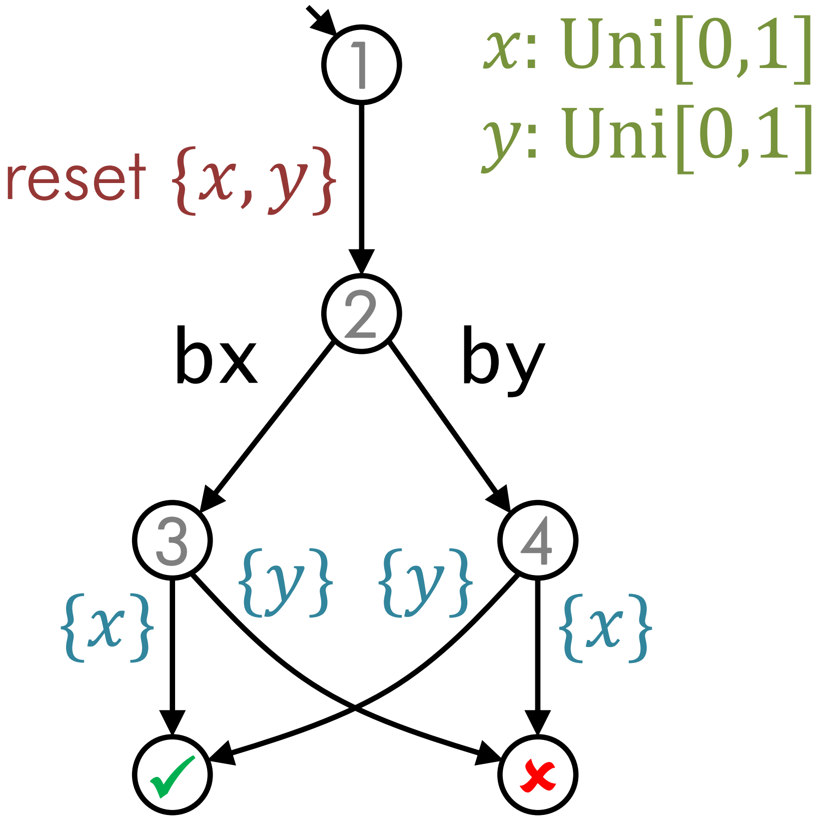
$$\to \langle 2, \langle 0,0\rangle, \langle 0.2,0.8\rangle \rangle \quad \text{Uni}[0,1]^2$$

$$\xrightarrow{\text{choice}} \langle 3, \langle 0,0\rangle, \langle 0.6,0.4\rangle \rangle \quad \text{resample } x, y$$

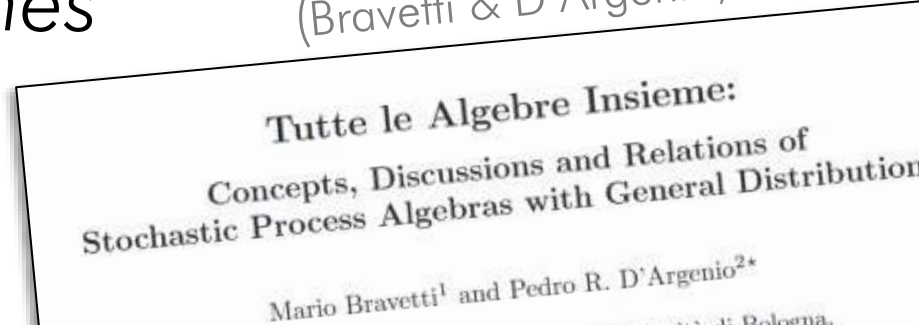
$$\to \langle 3, \langle 0.4,0.6\rangle, \langle 0.9,0.4\rangle \rangle \quad \text{resample } x$$

$$\to \langle ✗, \langle 0.2,0.2\rangle, \langle 0.2,0.8\rangle \rangle$$

→ *spent lifetimes*
semantics



(Bravetti & D'Argenio, 2004)

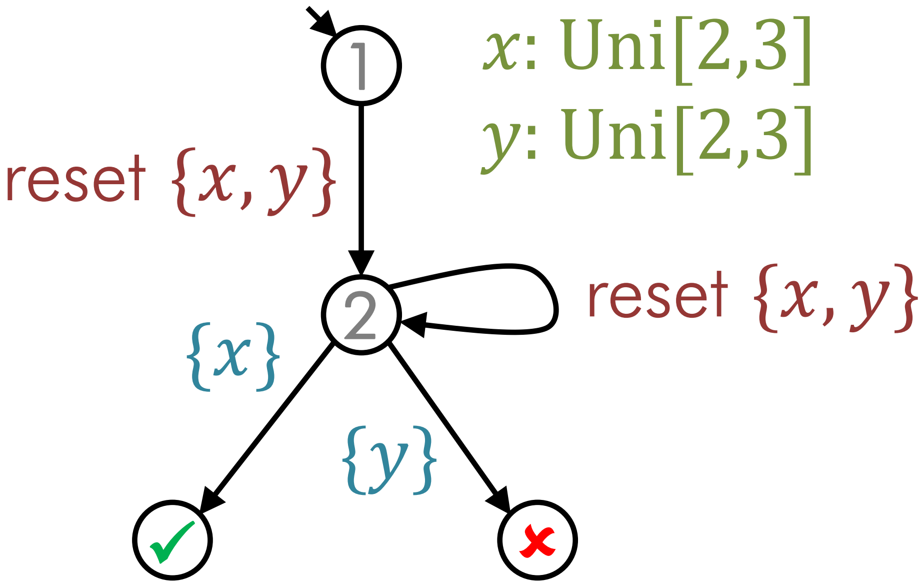


Tutte le Algebre Insieme:
Concepts, Discussions and Relations of Stochastic Process Algebras with General Distributio

Mario Bravetti[1] and Pedro R. D'Argenio[2*]

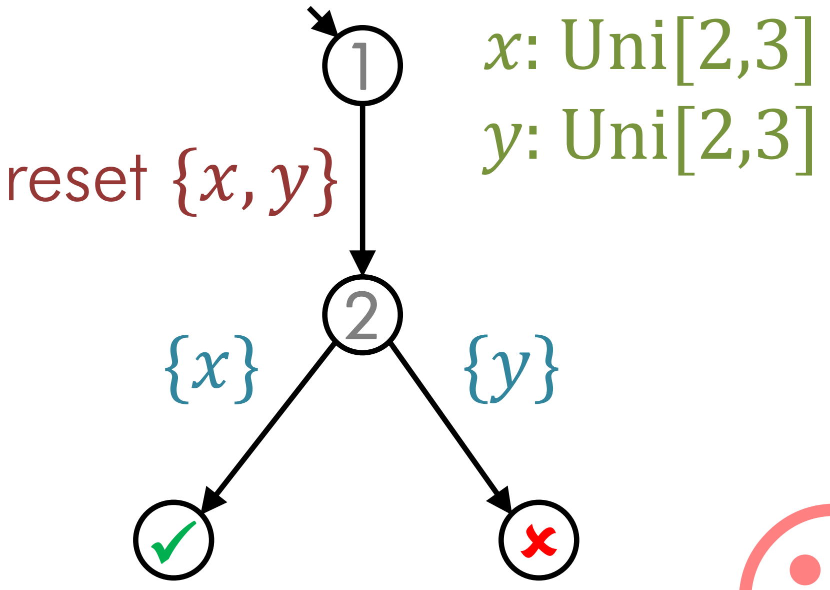
Fixing the Semantics… not.

Spent lifetimes still admits prophetic scheduling:



(pick loop until $x < y$)

Spent lifetimes *also* admits *divine scheduling*:

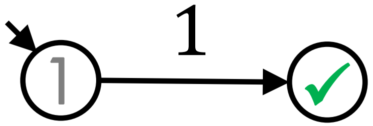


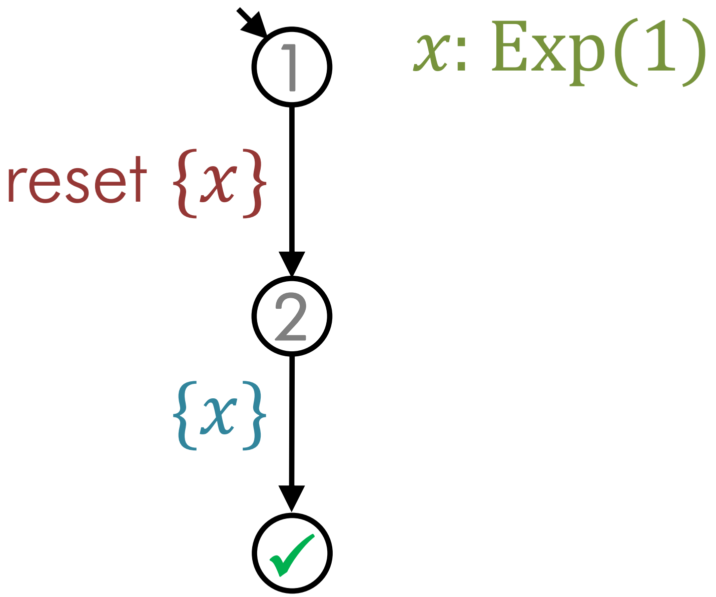
(delay $1, \frac{1}{2}, \frac{1}{4}, \dots$ until $x < y$)

Fixing the Semantics… not.

…and divine scheduling
is even worse:

as a CTMC:



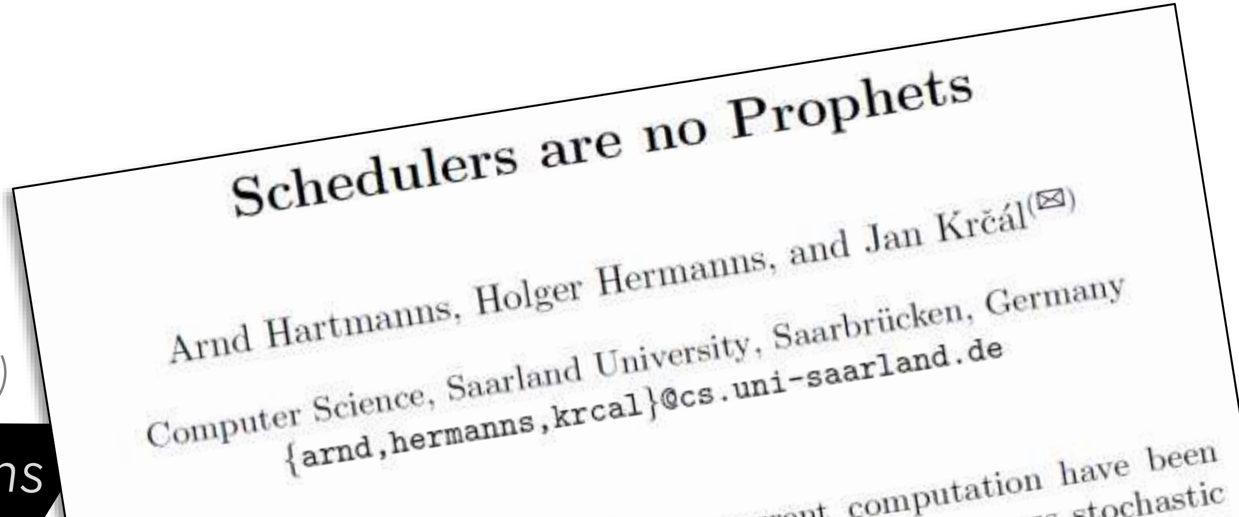


Expected: Reach ✓ after
1 time unit on average

Actually: Reach ✓ with
min. probability zero

(always delay
for $\frac{v(x)}{2}$ in 2)

(H., Hermanns, Krčál, Semantics, Logics, and Calculi, 2016)



Schedulers are no Prophets

Arnd Hartmanns, Holger Hermanns, and Jan Krčál(✉)

Computer Science, Saarland University, Saarbrücken, Germany
{arnd,hermanns,krcal}@cs.uni-saarland.de

computation have been
stochastic

Limiting Scheduler Power

What is
a useful class
of scheduler for
stochastic automata?

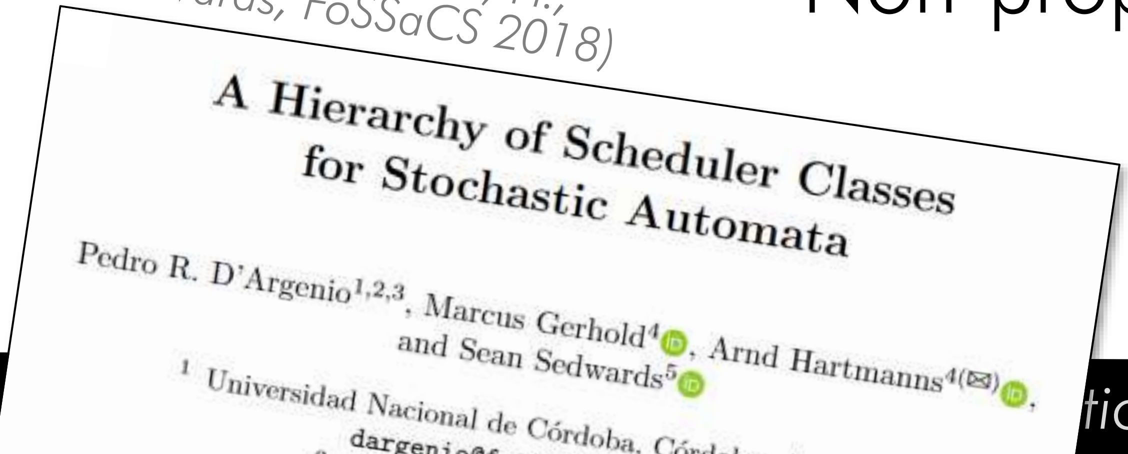
Classic:

$$\begin{array}{ccccccc}
\mathfrak{S}^{ml}_{\ell,o} & \prec & \mathfrak{S}^{ml}_{\ell,t,o} & \prec & \mathfrak{S}^{ml}_{\ell,v,o} & \prec & \mathfrak{S}^{hist}_{\ell,o} \\
\not\wr & & \not\wr & & \curlywedge & & \wr \\
\mathfrak{S}^{ml}_{\ell,e} & \prec & \mathfrak{S}^{ml}_{\ell,t,e} & \prec & \mathfrak{S}^{ml}_{\ell,v,e} & \succ & \mathfrak{S}^{hist}_{\ell,t,o} \\
\curlywedge & & \curlywedge & & \wr & & \wr \\
\mathfrak{S}^{hist}_{\ell,e} & \approx & \mathfrak{S}^{hist}_{\ell,t,e} & \approx & \mathfrak{S}^{hist}_{\ell,v,e} & \succ & \mathfrak{S}^{hist}_{\ell,v,o}
\end{array}$$

Non-prophetic:

$$\begin{array}{ccccc}
\mathfrak{S}^{ml}_{\ell} & \prec & \mathfrak{S}^{ml}_{\ell,t} & \prec & \mathfrak{S}^{ml}_{\ell,v} \\
\curlywedge & & \curlywedge & & \curlywedge \\
\mathfrak{S}^{hist}_{\ell} & \approx & \mathfrak{S}^{hist}_{\ell,t} & \approx & \mathfrak{S}^{hist}_{\ell,v}
\end{array}$$

(D'Argenio, Gerhold, H., Sedwards, FoSSaCS 2018)



A Hierarchy of Scheduler Classes for Stochastic Automata

Pedro R. D'Argenio[1,2,3], Marcus Gerhold[4], Arnd Hartmanns[4](✉), and Sean Sedwards[5]

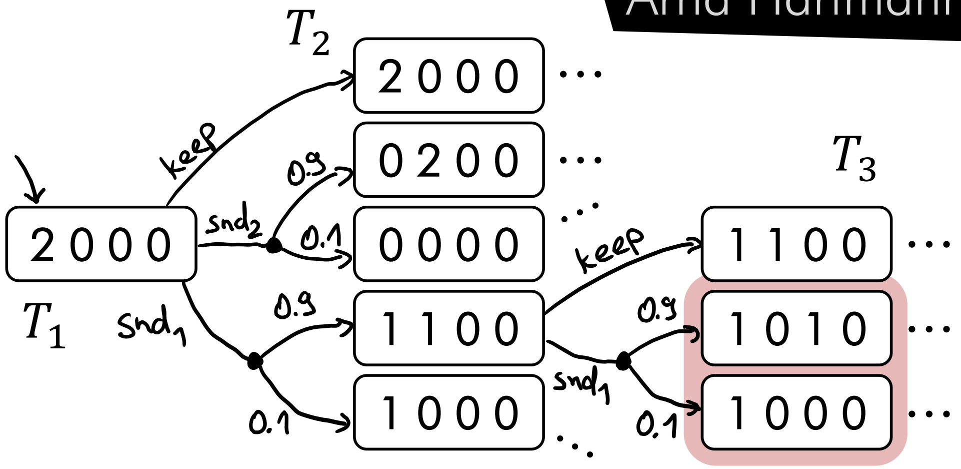
[1] Universidad Nacional de Córdoba, Córd

dargenio@

3 open problems

Arnd Hartmanns

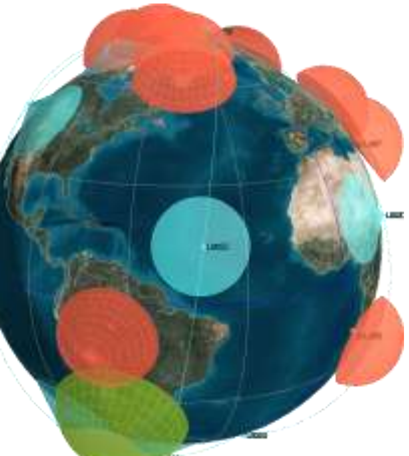
Schedulers and Information in Distributed and Non-Markovian Probabilistic Systems

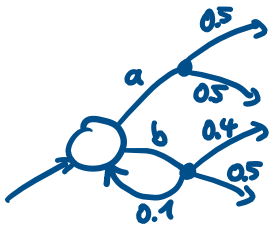


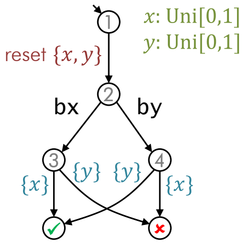
Need to find **useful** tradeoffs between

model expressiveness & scheduling power | tractability of analysis & usefulness of schedulers

in non-trivial probabilistic and stochastic timed settings







…per modelling formalism?

…per application scenario?