Simulation-based Verification for Stochastic Systems

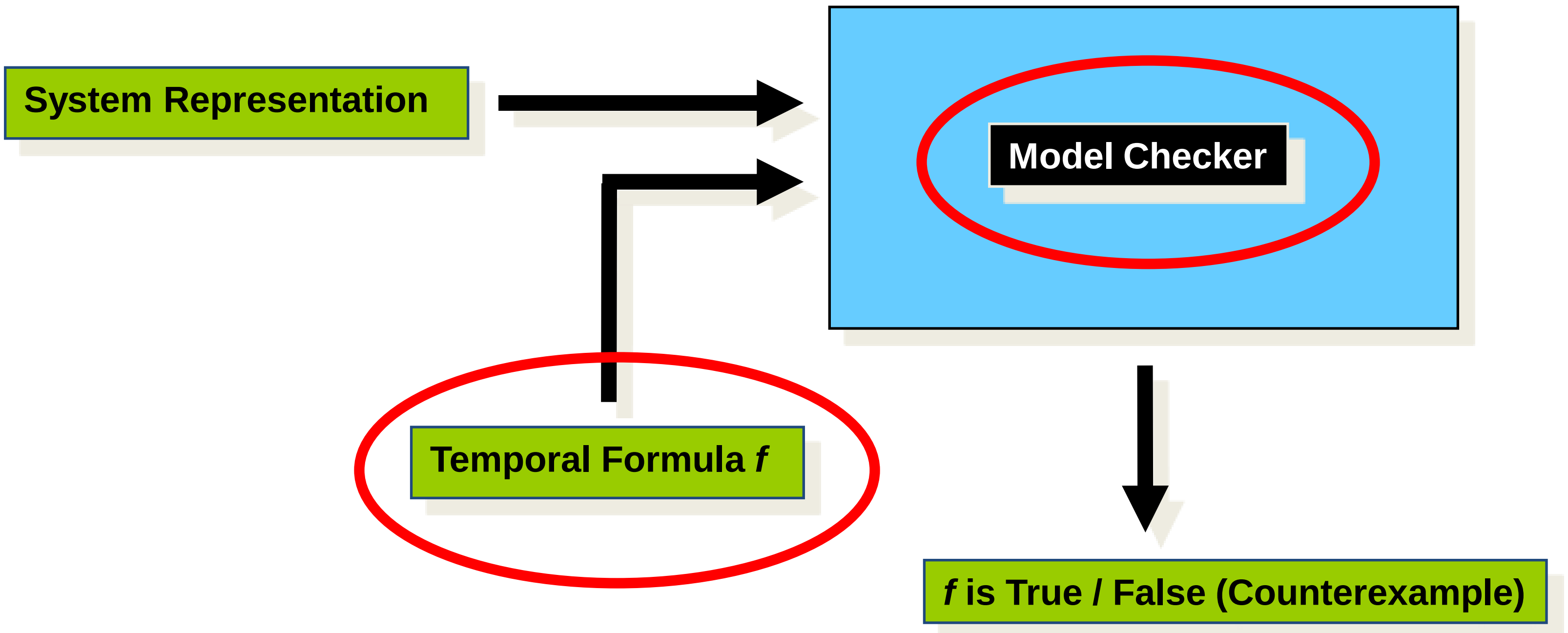
Paolo Zuliani

Outline

- Formal verification and Model Checking
- Part I: Markov Decision Processes
- Part II: Stochastic Hybrid Systems

Formal Verification

"Making (mathematically) **sure** that systems *perform as expected"*



Real-World Impact

- Pnueli received the 1996 *Turing Award* (‘*Nobel* prize of computing’) for temporal logic
- Clarke, Emerson, and Sifakis received the 2007 *Turing Award* for (temporal logic) model checking
- Massive real-world impact:
  - most hardware designs (*e.g.*, smartphone chips) are now *formally verified*
  - used by large companies (*Intel*, *Microsoft*, *etc.*)

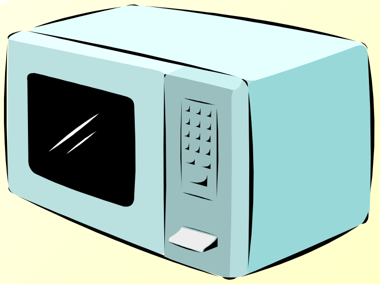
Temporal Logic

A formal (mathematical) notation to express *temporal relations* between events

For example, a microwave oven should satisfy:

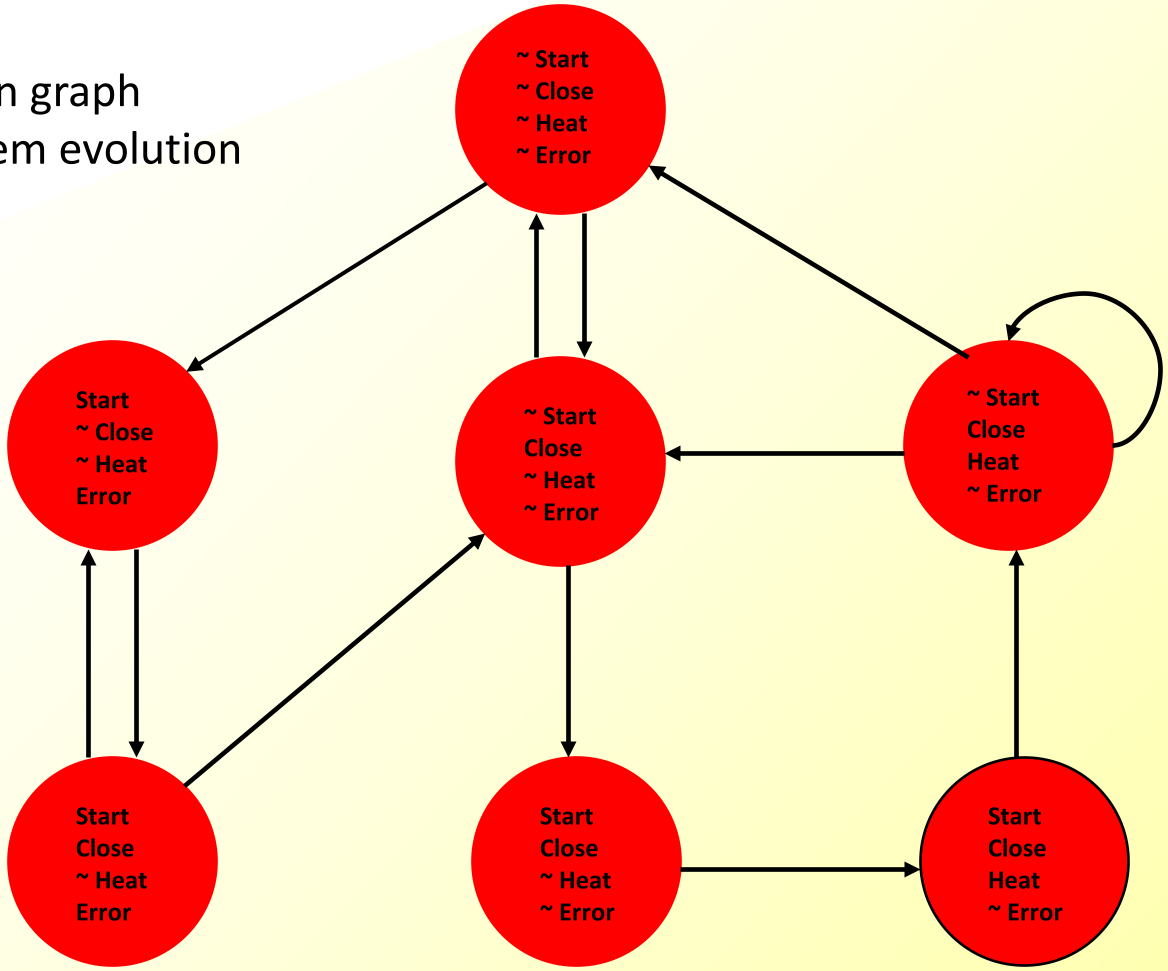
- The oven doesn't heat up until the door is closed
- Not heat_up holds until door_closed
- (~ heat_up) U door_closed
- A ((~ heat_up) U door_closed)

Model Checking

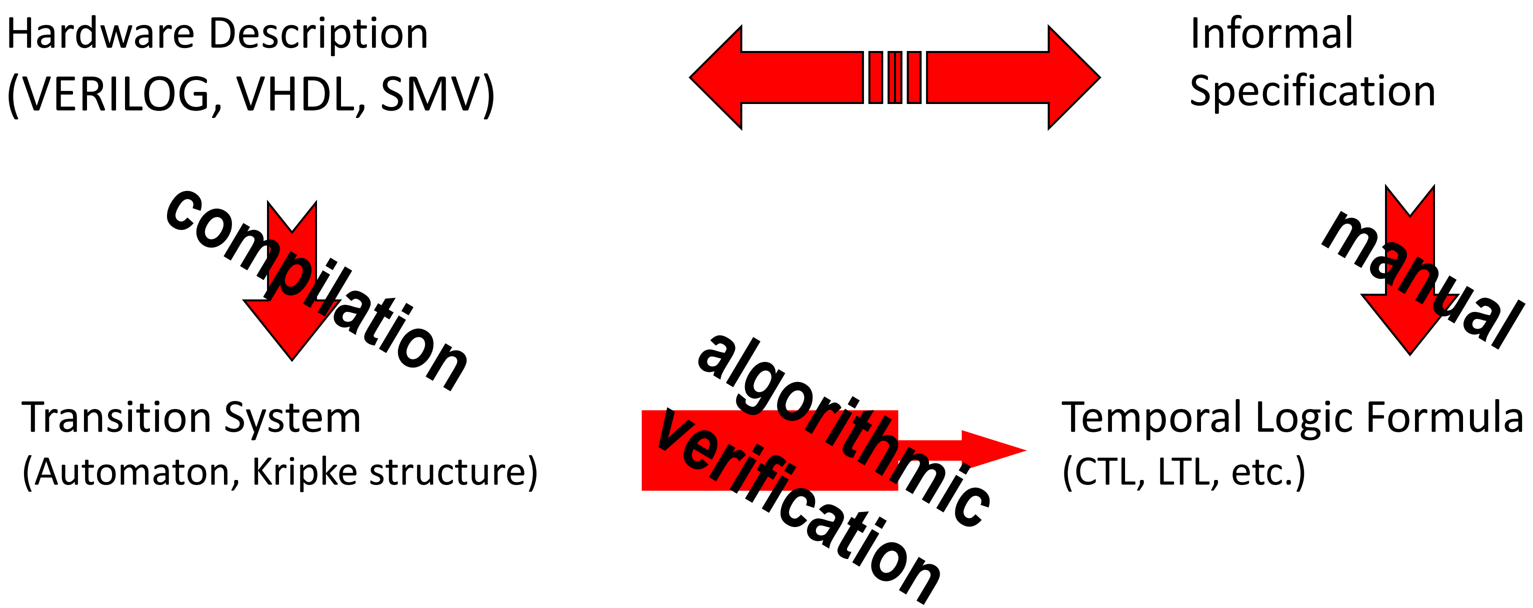


An **intelligent exhaustive search of the state space** of the design

State-transition graph describes system evolution over time



Model Checking



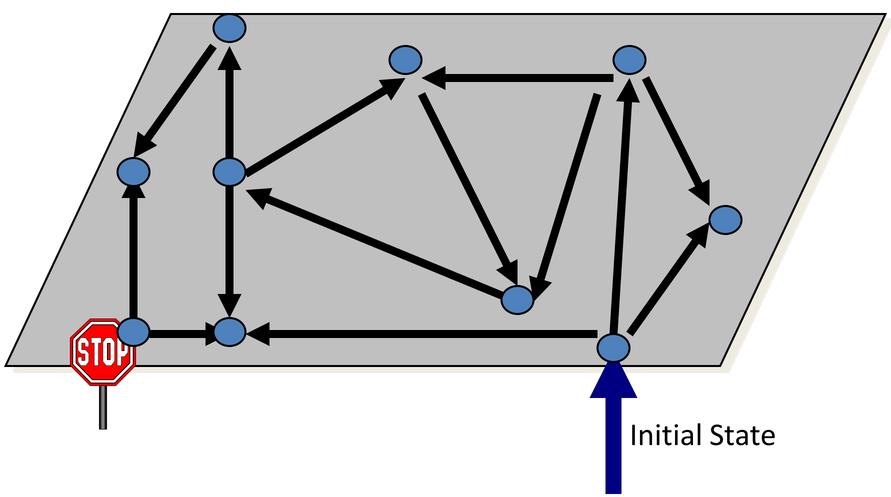
Counterexamples

Program or circuit

Informal Specification

Transition System

Temporal Logic Formula
(CTL, LTL, etc.)



Safety Property:
Is bad state unreachable:

satisfied

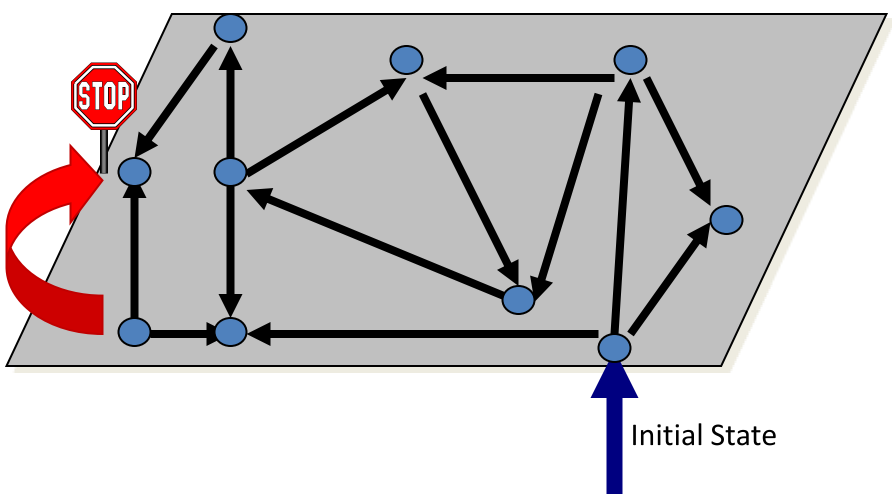
Counterexamples

Program or circuit

Informal Specification

Transition System

Temporal Logic Formula
(CTL, LTL, etc.)



Safety Property:
bad state STOP unreachable

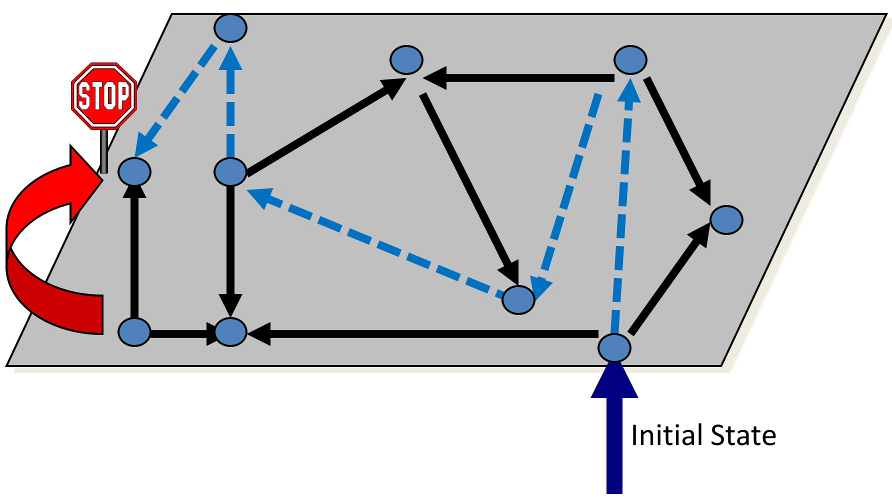
Counterexamples

Program or circuit

Informal Specification

Transition System

Temporal Logic Formula
(CTL, LTL, etc.)



Safety Property:
bad state STOP unreachable

Counterexample

Verification of Stochastic Models

- Temporal properties over the model's (stochastic) evolution

- For a property Φ and a fixed $0<\vartheta<1$, we ask whether

$$P_{\geq\vartheta}(\Phi) \quad \text{or} \quad P_{<\vartheta}(\Phi)$$

- For example: "does GFP reach 4,000 within 20 minutes, with probability at least 0.99?"

Simulation-based Verification

- State Space Exploration infeasible for large systems
  - Symbolic MC with OBDDs can address large state spaces
  - But scalability depends on the structure of the system
- Pros: simulation is feasible for **many more** systems
  - Often easier to simulate a complex system than to build the transition relation for it
- Pros: easier to parallelize
- Cons: answers may be **wrong**
  - But error probability can be bounded
- Cons: simulation is **incomplete** (continuous state spaces)

Statistical Model Checking

Key idea

(Haakan Younes, 2001)

- Suppose system behavior w.r.t. a (fixed) property *Φ* can be modeled by a Bernoulli of parameter *p*:
  - System satisfies *Φ* with (unknown) probability *p*
- Questions: $P_{\geq\vartheta}(\Phi)$? (for a fixed $0<\vartheta<1$)
- Draw a sample of system simulations and use:
  - Statistical hypothesis testing: Null vs. Alternative hypothesis

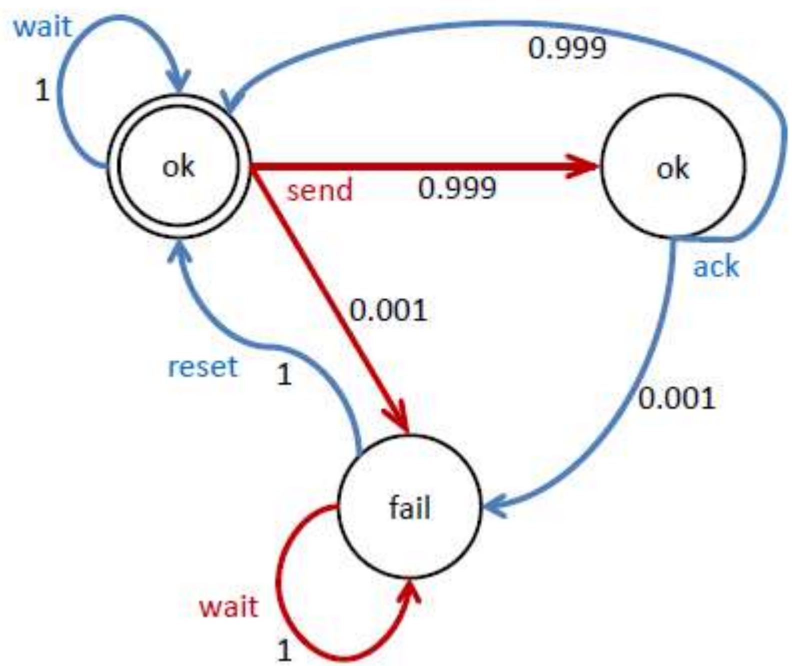
$$H_0 : \mathcal{M} \models P_{\geqslant\theta}(\phi) \qquad H_1 : \mathcal{M} \models P_{<\theta}(\phi)$$

  - Statistical estimation: returns "*p* in (a,b)" (and compare a with ϑ)

Nondeterministic Systems

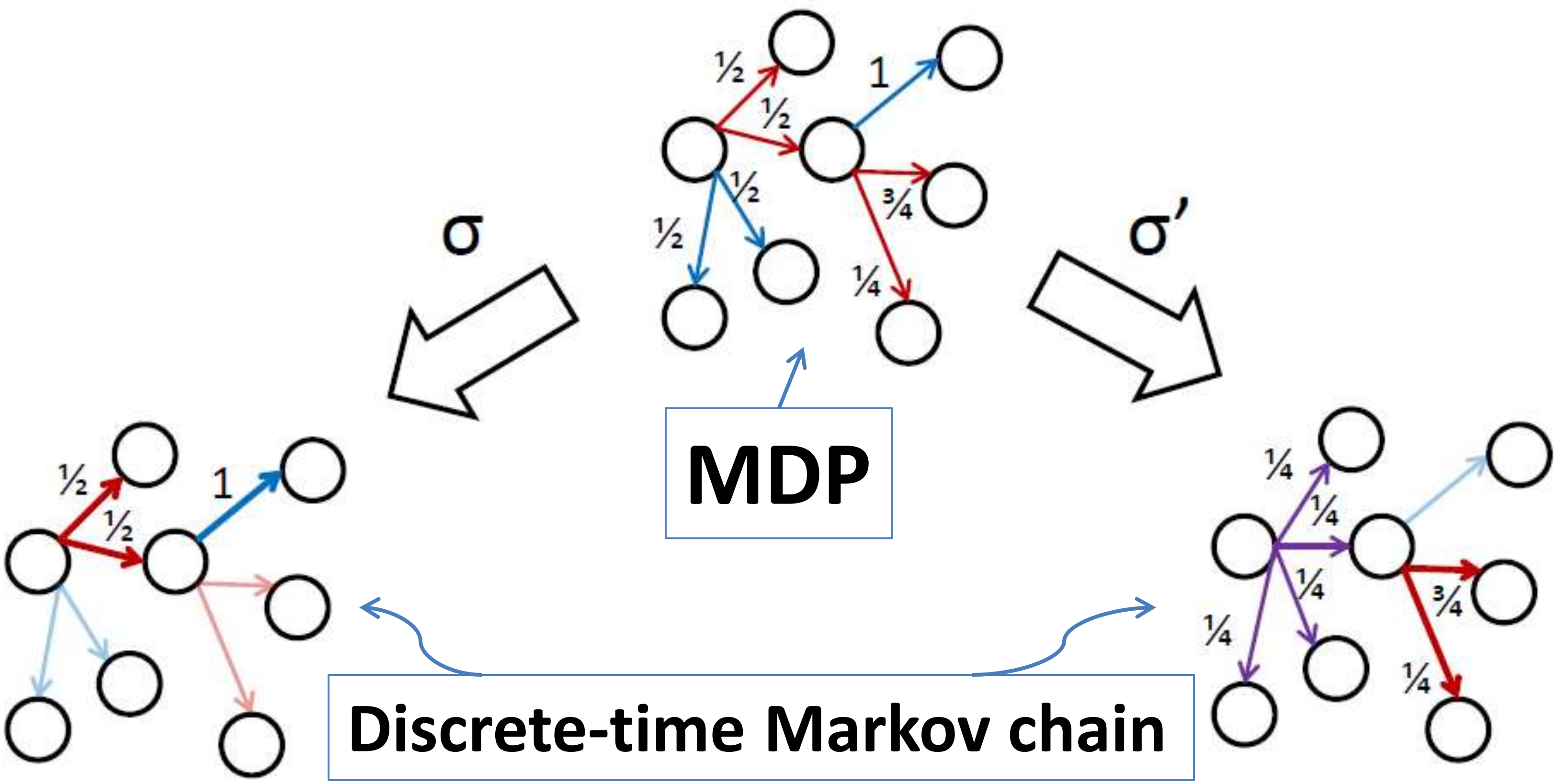
- Problem: sampling-based methods have no way to choose which pure non-deterministic action or outcome to follow when creating a sample execution trace.

Markov Decision Processes (MDPs)



Resolving Nondeterminism

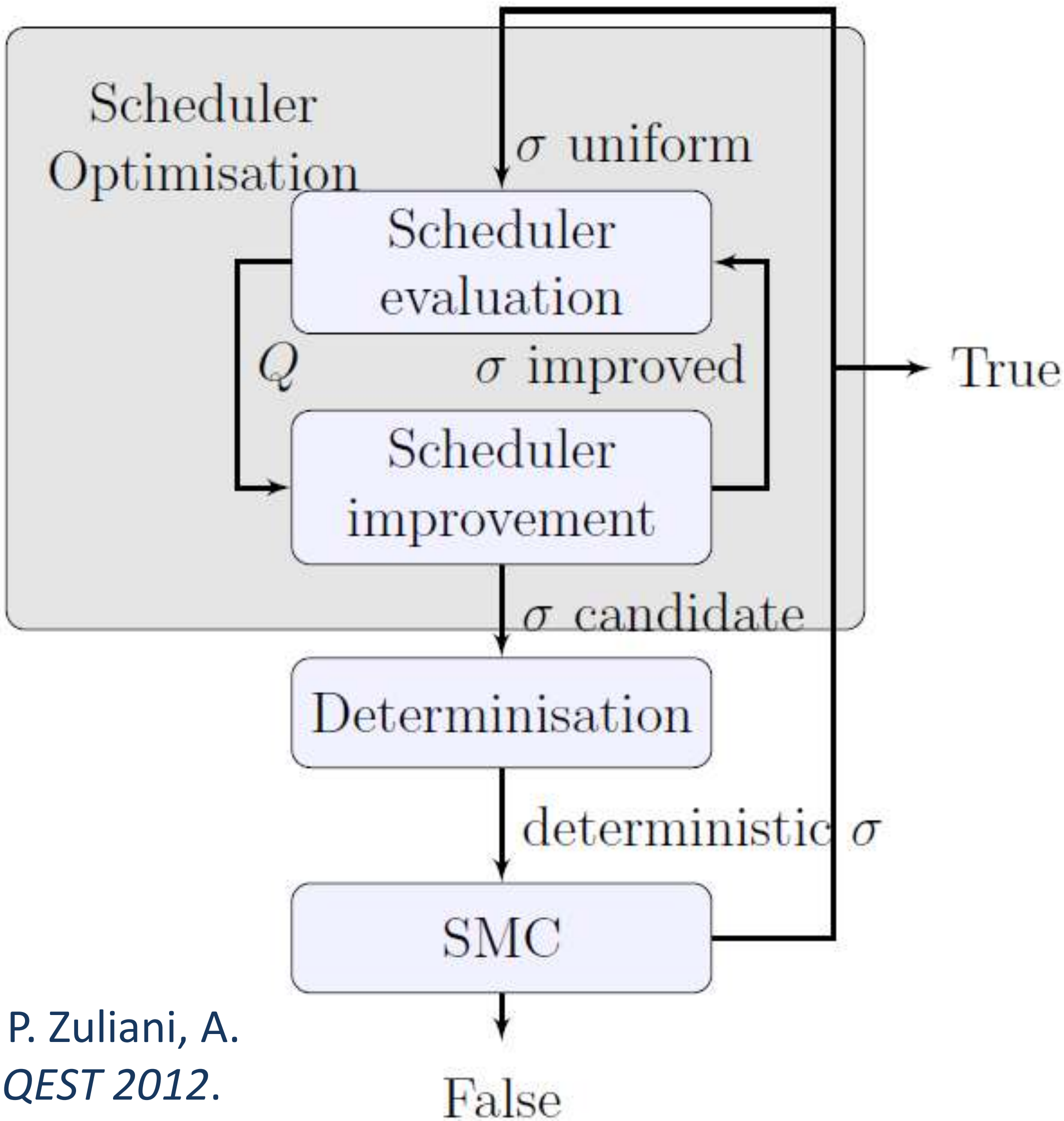
- Memory-less stochastic policy or "scheduler" can resolve nondeterminism.
- Specifies choices in each state:



Nondeterministic Systems

- Different resolution of nondeterminism (schedulers) can result in different behaviors
- Max and min probability that a property Φ holds
- Question: is $Prob(\Phi) \leq \vartheta$, for ***all schedulers***?
- How to find the optimal scheduler:
  - maximizes (minimizes) probability that Φ holds

Our Approach



D. Henriques, J. Martins, P. Zuliani, A. Platzer and E. M. Clarke. *QEST 2012*.

Bounded Linear Temporal Logic

- Bounded Linear Temporal Logic (BLTL): A version of LTL with time bounds on temporal operators.
- Let $\sigma = (s_0, t_0), (s_1, t_1), \ldots$ be an execution of the model
  - along states $s_0, s_1, \ldots$
  - the system stays in state s_i *for time* t_i
  - divergence of time: $\Sigma_i\, t_i$ diverges (i.e., non-zeno)
- σ^i: Execution trace starting at state *i*
- A model for simulation traces

BLTL: Examples

- *"within 600 time units, the number of p53 molecules will be greater than 900"*

$$\boxed{\mathbf{F}^{600}\ (\ p53 > 900\ )}$$

- *"within 200 time units, p53 will stay below 33,000 for 900 time units"*

$$\boxed{\mathbf{F}^{200}\ (\ \mathbf{G}^{900}\ (\ p53 < 3.3 \times 10^{4}\ )\ )}$$

- *"within 100 t.u., p53 will pass 2,000, and in the next 100 t.u. it will eventually be below 1,000"*

$$\boxed{\mathbf{F}^{100}\ (\ \mathrm{p53} \geq 2{,}000\ \ \&\ \mathbf{F}^{100}\ (\ \mathrm{p53} \leq 1{,}000\ )\ )}$$

Semantics of BLTL

The semantics of BLTL for a trace σ^k:

- $\sigma^k \models AP$ iff atomic proposition AP true in state s_k
- $\sigma^k \models \Phi_1 \vee \Phi_2$ iff $\sigma^k \models \Phi_1$ or $\sigma^k \models \Phi_2$
- $\sigma^k \models \neg\Phi$ iff $\sigma^k \models \Phi$ does not hold
- $\sigma^k \models \Phi_1 \, \mathcal{U}^t \, \Phi_2$ iff there exists natural i such that
  1) $\sigma^{k+i} \models \Phi_2$
  2) $\Sigma_{j<i} \, t_{k+j} \leq t$
  3) for each $0 \leq j < i$, $\sigma^{k+j} \models \Phi_1$

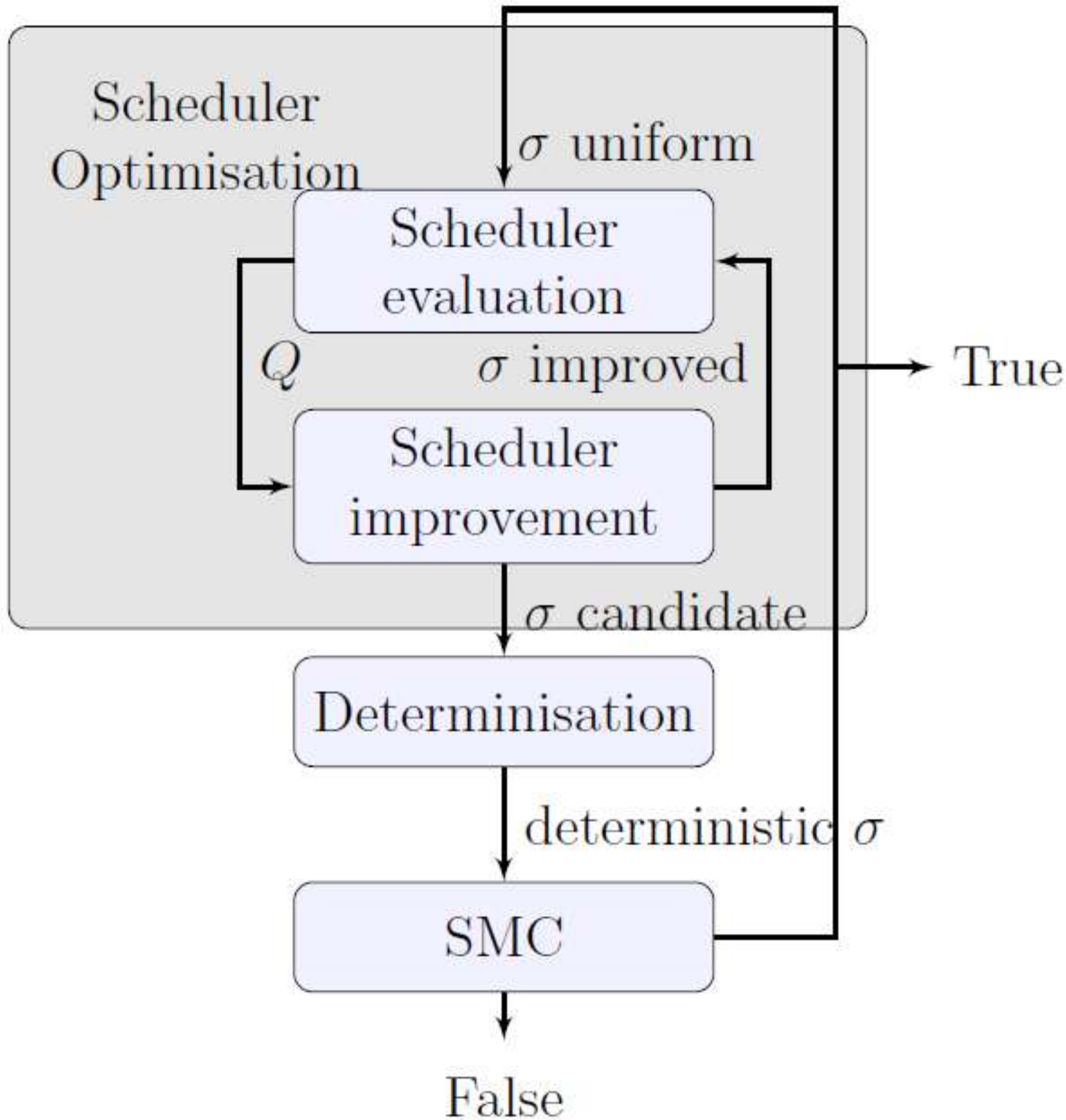
  "within time t, Φ_2 will be true and Φ_1 will hold until then"

- In particular, $F^t \Phi = true \, \mathcal{U}^t \, \Phi$, $G^t \Phi = \neg F^t \neg\Phi$

SMC for Markov Decision Processes

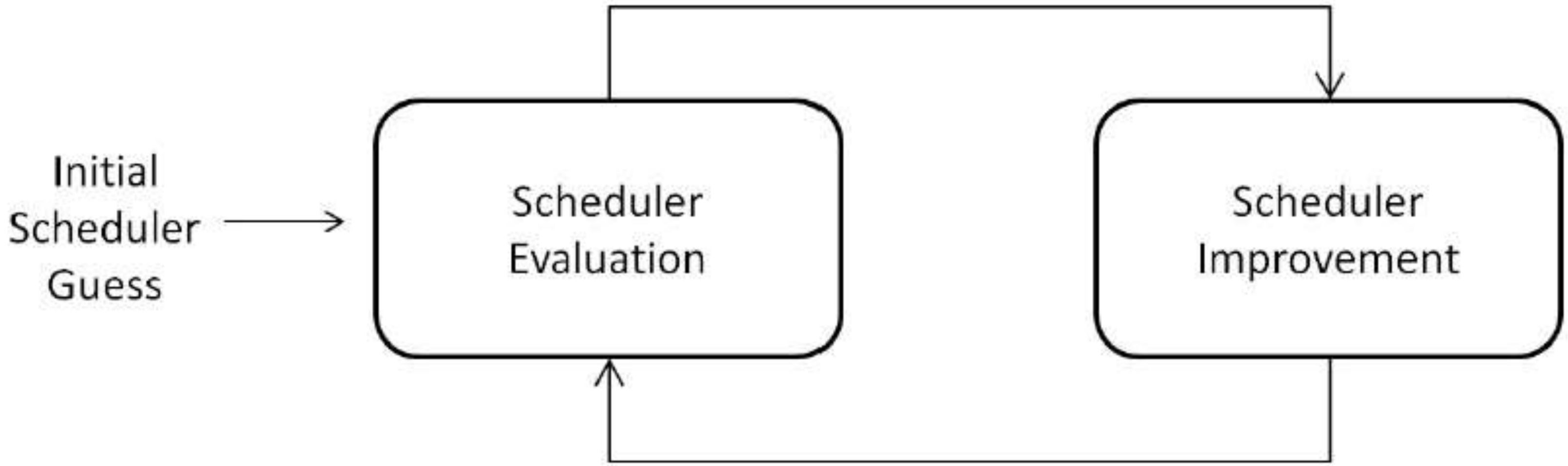
- A guided search for the optimal scheduler using reinforcement learning:
  - Simulate the system keeping track of the transitions taken, and check property Φ
  - Reinforce the "good" transitions (*i.e.*, those leading to property satisfaction)
- Recall that: MDP + scheduler = DTMC

Our Approach



Scheduler Evaluation & Improvement

- Learn the most adversarial choices at each state, by successively refining an initial guess.



- ***Reinforcement learning***, where quality is based on how often state/action choices occur in traces that satisfy the property in question.

Scheduler Evaluation & Improvement

- Quality $Q_\sigma(s, a)$ of state s, action a is

$Prob_\sigma$(traces satisfying Φ and containing (s, a))

- Scheduler evaluation:
  - $Q_\sigma(s, a)$ is estimated via simulation
- Scheduler improvement:
  - Give more probability to transitions with higher quality (*i.e.*, higher $Q_\sigma(s, a)$)

Scheduler Evaluation & Improvement

- Quality $Q_\sigma(s, a)$ is estimated via finite sample-size simulation:

$$\widehat{Q_\sigma}(s,a) = \frac{\#\{\pi \mid \pi \vdash \phi \ \wedge (s,a) \in \pi\}}{\#\{\pi \mid (s,a) \in \pi\}}$$

- Improving a scheduler σ:

$$\sigma'(s,a) = \frac{\widehat{Q_\sigma}(s,a)}{\sum_\alpha \widehat{Q_\sigma}(s,\alpha)}$$

More details in our QEST 2012 paper…

Convergence

- Value of a state under a scheduler:

$$V_\sigma(s) = Prob_\sigma(\pi \mid \pi \vdash \phi \wedge (s, a) \in \pi \wedge a \in \mathrm{A}(s))$$

- Note that:

$$Prob_\sigma(\pi \mid \pi \vdash \phi) = V_\sigma(\bar{s}) = \sum_{a \in \mathrm{A}(\bar{s})} \sigma(\bar{s}, a) \cdot Q_\sigma\,(\bar{s}, a)$$

Convergence

- We show that if σ is a scheduler and σ' is our improved scheduler, then:

$$V_{\sigma'}(\bar{s}) \geq V_{\sigma}(\bar{s})$$

- But we might converge to a local optimum …

Correctness

- **Question**: is $Prob_\sigma(\Phi) \leq \vartheta$, for all schedulers σ?
- If we find a scheduler σ such that

$$Prob_\sigma(\Phi) > \vartheta$$

  then we are done. The answer is 'no' and we can trust it.
- Otherwise:
  - The question above may be true; or
  - We ended up in a local optimum
- We restart the algorithm to exponentially increase confidence in answer 'yes'

SMC for Markov Decision Processes

- Parallel implementation in Prism
- Can be faster than Prism on some problems
- Can provide *counterexample* schedulers

Experiments: Network protocols

CSMA 3 4							
	θ	0.5	0.8	**0.85**	**0.9**	0.95	**PRISM**
	out	F	F	F	T	T	**0.86**
	t	1.7	11.5	35.9	115.7	111.9	136

CSMA 3 6							
	θ	0.3	0.4	**0.45**	**0.5**	0.8	**PRISM**
	out	F	F	F	T	T	**0.48**
	t	2.5	9.4	18.8	133.9	119.3	2995

CSMA 4 4							
	θ	0.5	0.7	0.8	**0.9**	**0.95**	**PRISM**
	out	F	F	F	F	T	**0.93**
	t	3.5	3.7	17.5	69.0	232.8	16244

CSMA 4 6							
	θ	0.5	0.7	0.8	0.9	**0.95**	**PRISM**
	out	F	F	F	F	F	timeout
	t	3.7	4.1	4.2	26.2	258.9	timeout

WLAN 5							
	θ	0.1	**0.15**	**0.2**	0.25	0.5	**PRISM**
	out	F	F	T	T	T	**0.18**
	t	4.9	11.1	124.7	104.7	103.2	1.6

WLAN 6							
	θ	0.1	**0.15**	**0.2**	0.25	0.5	**PRISM**
	out	F	F	T	T	T	**0.18**
	t	5.0	11.3	127.0	104.9	102.9	1.6

Experiments: Two robots

- *n* by *n* grid
- Robot movements are imprecise (*r* = scattering radius)

Robot $n = 50$ $r = 1$	$\boldsymbol{\theta}$	0.9	0.95	**0.99**	**PRISM**
	out	F	F	F	**0.999**
	t	23.4	27.5	40.8	1252.7

Robot $n = 50$ $r = 2$	$\boldsymbol{\theta}$	0.9	0.95	**0.99**	**PRISM**
	out	F	F	F	**0.999**
	t	71.7	73.9	250.4	3651.045

Robot $n = 75$ $r = 2$	$\boldsymbol{\theta}$	0.95	0.97	**0.99**	**PRISM**
	out	F	F	F	timeout
	t	382.5	377.1	2676.9	timeout

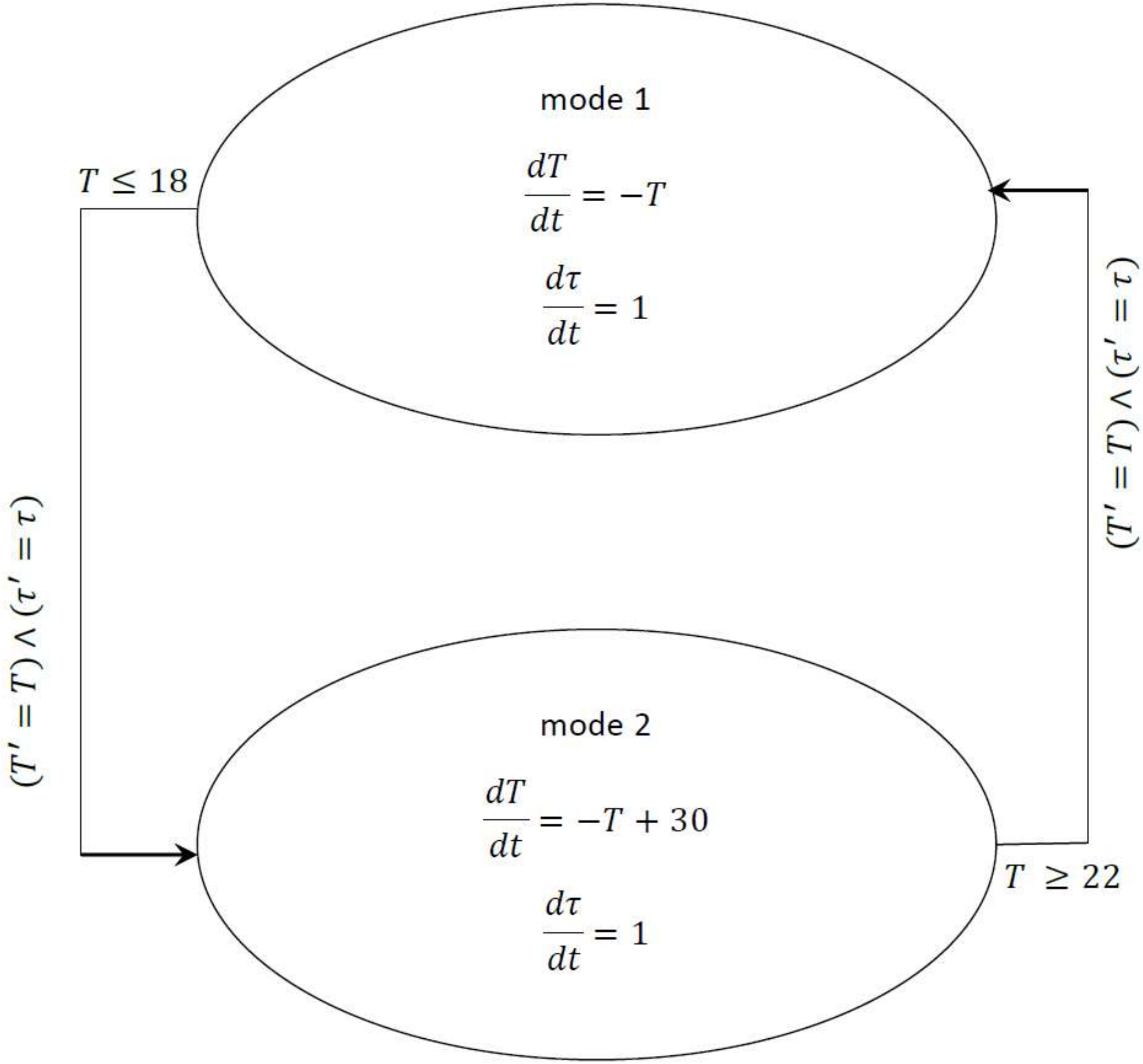
Robot $n = 200$ $r = 3$	$\boldsymbol{\theta}$	0.85	**0.9**	**0.95**	**PRISM**
	out	F	F	T	timeout
	t	903.1	1129.3	2302.8	timeout

Conclusions (Part I)

- Simulation-based verification of MDP is:
  - Possible!
  - Efficient (better than Prism in some cases)
- Possible extensions:
  - Unbounded properties, general schedulers, CTMDP (?), *etc*.

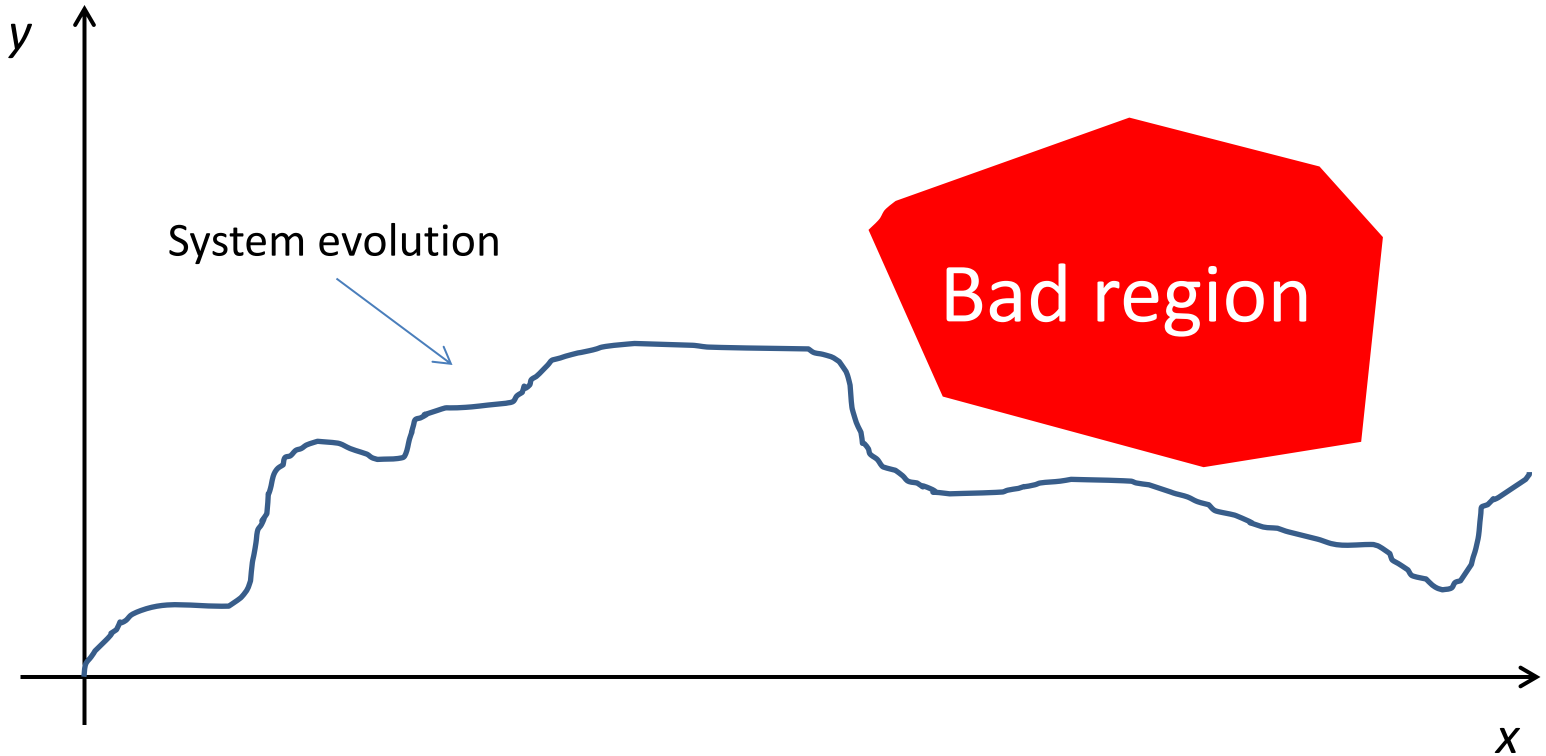
Part II: Stochastic Hybrid Systems

- Hybrid System:
  - Combine continuous and discrete evolution
  - A model for cyber-physical systems



Reachability

- Reachability properties:
  - Does the system reach the bad region?



A Step Back

- Reachability is undecidable* even for linear (differential) hybrid systems!!
- So, the question is too hard for a computer, and we need to "relax" it
  - We need to reformulate the reachability problem into an easier one

*It is **impossible** to develop an algorithm that for any hybrid system and region will tell us whether the system evolution reaches the region

δ-Reachability

- δ-reachability (Gao, Avigad, Clarke 2012) is instead decidable
- For $\delta > 0$, the system evolution may:

δ-satisfiable

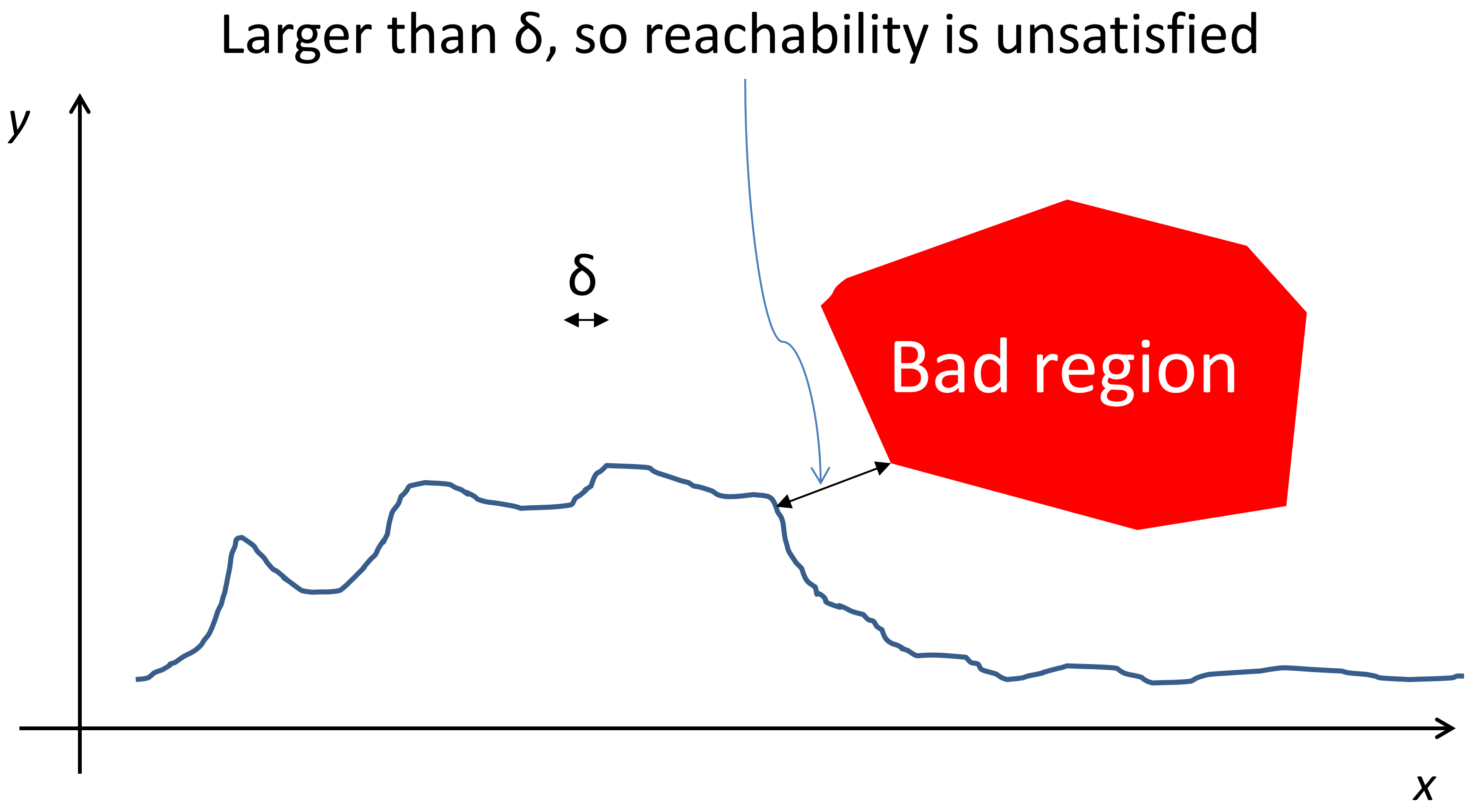
1. Get to a distance $< \delta$ from the bad region, without entering it
2. Enter the bad region

unsatisfiable

3. Stay out of the bad region (more than δ)

An algorithm solving the problem above is called ***δ-complete***

δ-Reachability



δ-Reachability

Smaller than δ, so reachability is δ-satisfiable

Stochastic Hybrid Systems

- We study Hybrid Systems with random initial parameters (US Navy grant with Clarke)
- E.g.: the initial temperature in the thermostat model is, say, normally distributed (Gaussian)
- Question: what is the probability that the temperature reaches 20C within 10mins?

F. Shmarov, P. Zuliani. *2014*.

Probabilistic δ-Reachability

- We want a δ-complete procedure for SHS with random initial parameters
- This boils down to computing integrals with verified results:
  - the integration algorithm returns an interval (size < δ) which is guaranteed to contain the true result
  - based on ***verified simulation*** algorithms for solving ODEs (computing interval enclosures)

Probabilistic δ-Reachability

Thermostat model ($\delta=10^{-9}$):

#	k	τ	Probability interval	CPU
1	1	0.6	[0.006693073099383227, 0.006693073733195108]	31
2	5	1.8	[0.002635117907540255, 0.002635118445341895]	188
3	7	2.4	[0.00160257761701815, 0.001602578290160313]	413

k = number of discrete transitions, τ = global time, CPU = CPU time in seconds

Probabilistic δ-Reachability

Thermostat model with 4 modes ($\delta=10^{-9}$):

#	k	τ	Probability interval	CPU
1	2	0.6	[0.0007687433606520627, 0.0007687433607436878]	53
2	6	1.7	[9.585015171225825e-08, 9.684797129694618e-08]	343
3	6	1.8	[0.003967491767795972, 0.003967492552568959]	708

k = number of discrete transitions, τ = global time, CPU = CPU time in seconds

Next Steps

- SHS with random initial parameters and nondeterministic parameters
- Allow stochastic differential equations in the modes
- Curtis has written a SBML->SMT2 translator
  - Parameter estimation for ODE models
  - Synbio design: pruning out unfeasible models
- For papers, tools, *etc.* please see my homepage