Progress on Parameter Synthesis for Markov Models

Joost-Pieter Katoen

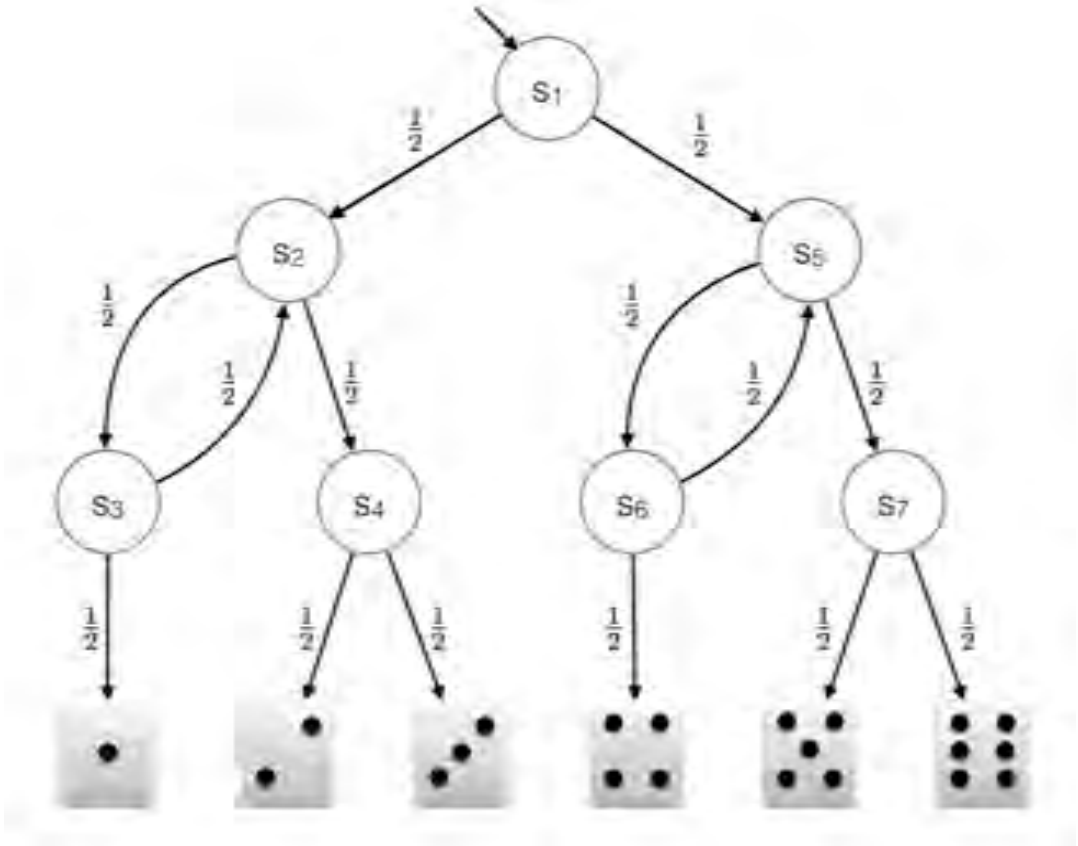


UNIVERSITY OF TWENTE.

Joint with: Christian Dehnert, Nils Jansen, Sebastian Junges, Tim Quatman, Erika Ábrahám, Harold Bruintjes, Florian Corzilius, Ufuk Topcu, Murat Cubutceke, Ivan Papusha, Hasan Poonawala, Matthias Volk

SYNCOP +PV'17 Workshop@Uppsala

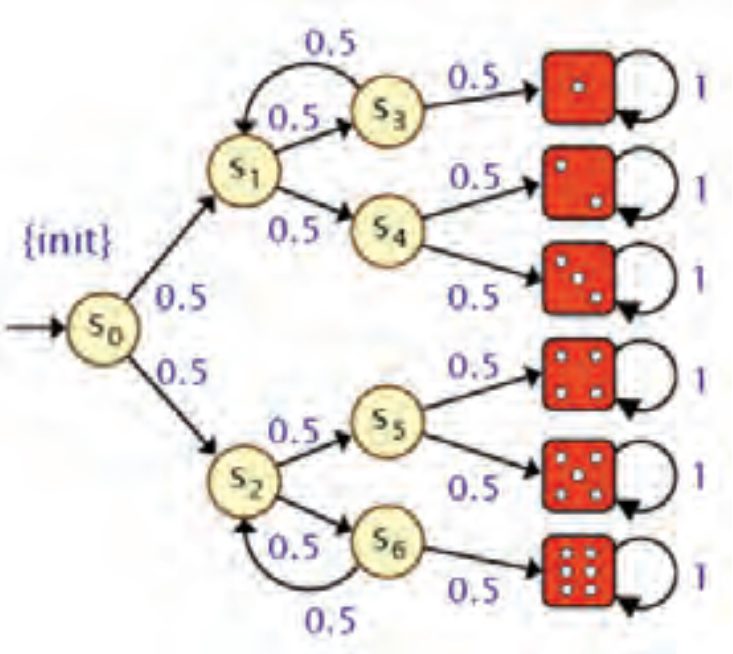
Randomised Algorithms: Simulating a Die [Knuth & Yao, 1976]



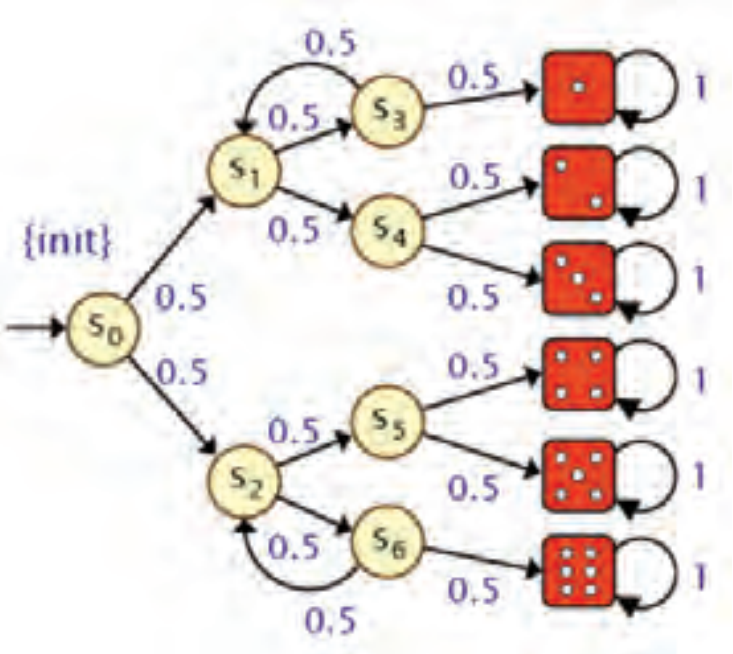
Heads = "go left"; tails = "go right". Does this model a six-sided die?

Reachability Probabilities: Knuth-Yao's Die

- Consider the event $\Diamond 4$



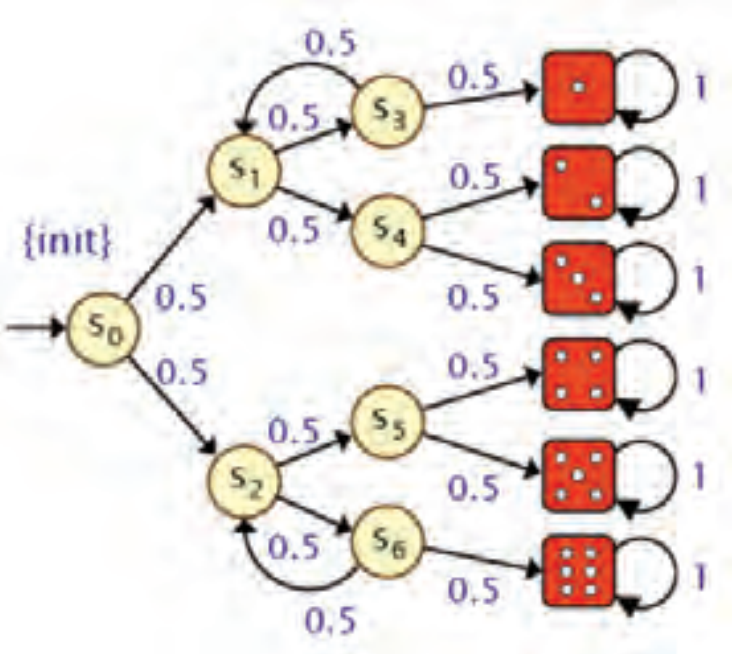
Reachability Probabilities: Knuth-Yao's Die



- Consider the event $\Diamond 4$
- We obtain:

$$x_1 = x_2 = x_3 = x_5 = x_6 = 0 \text{ and } x_4 = 1$$

Reachability Probabilities: Knuth-Yao's Die

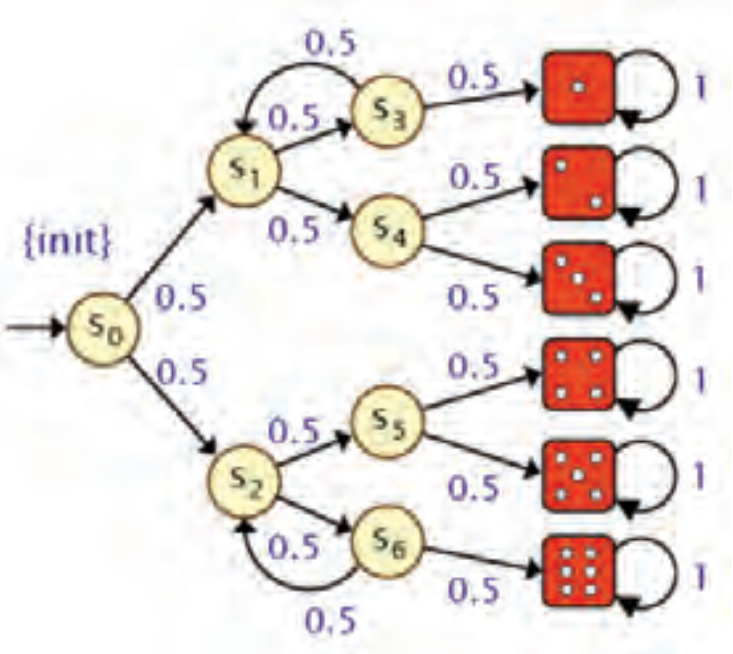


- Consider the event $\Diamond 4$
- We obtain:

$$x_1 = x_2 = x_3 = x_5 = x_6 = 0 \text{ and } x_4 = 1$$

$$x_{s_1} = x_{s_3} = x_{s_4} = 0$$

Reachability Probabilities: Knuth-Yao's Die



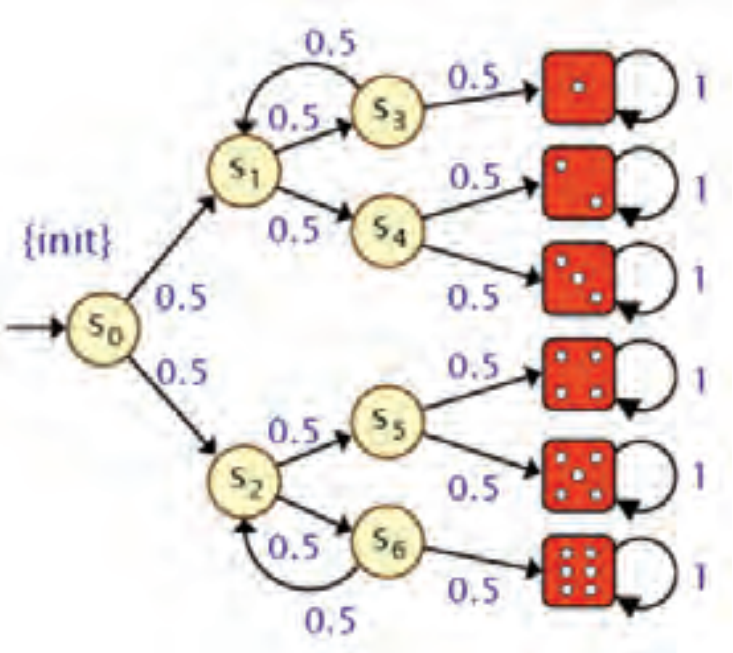
- Consider the event $\Diamond 4$
- We obtain:

$$x_1 = x_2 = x_3 = x_5 = x_6 = 0 \text{ and } x_4 = 1$$

$$x_{s_1} = x_{s_3} = x_{s_4} = 0$$

$$x_{s_0} = \frac{1}{2}x_{s_1} + \frac{1}{2}x_{s_2}$$

Reachability Probabilities: Knuth-Yao's Die



- Consider the event $\Diamond 4$
- We obtain:

$$x_1 = x_2 = x_3 = x_5 = x_6 = 0 \text{ and } x_4 = 1$$

$$x_{s_1} = x_{s_3} = x_{s_4} = 0$$

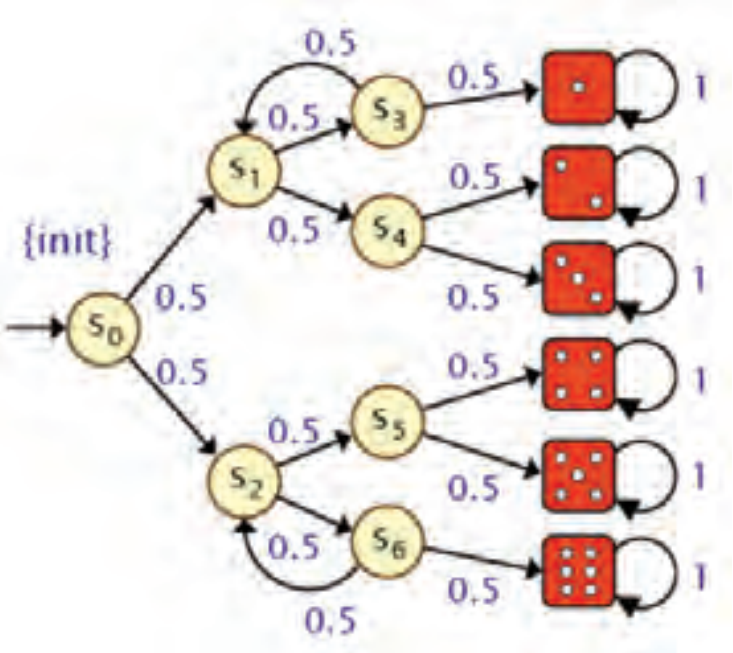
$$x_{s_0} = \frac{1}{2}x_{s_1} + \frac{1}{2}x_{s_2}$$

$$x_{s_2} = \frac{1}{2}x_{s_5} + \frac{1}{2}x_{s_6}$$

$$x_{s_5} = \frac{1}{2}x_5 + \frac{1}{2}x_4$$

$$x_{s_6} = \frac{1}{2}x_{s_2} + \frac{1}{2}x_6$$

Reachability Probabilities: Knuth-Yao's Die



- Consider the event $\Diamond 4$
- We obtain:

$$x_1 = x_2 = x_3 = x_5 = x_6 = 0 \text{ and } x_4 = 1$$

$$x_{s_1} = x_{s_3} = x_{s_4} = 0$$

$$x_{s_0} = \frac{1}{2}x_{s_1} + \frac{1}{2}x_{s_2}$$

$$x_{s_2} = \frac{1}{2}x_{s_5} + \frac{1}{2}x_{s_6}$$

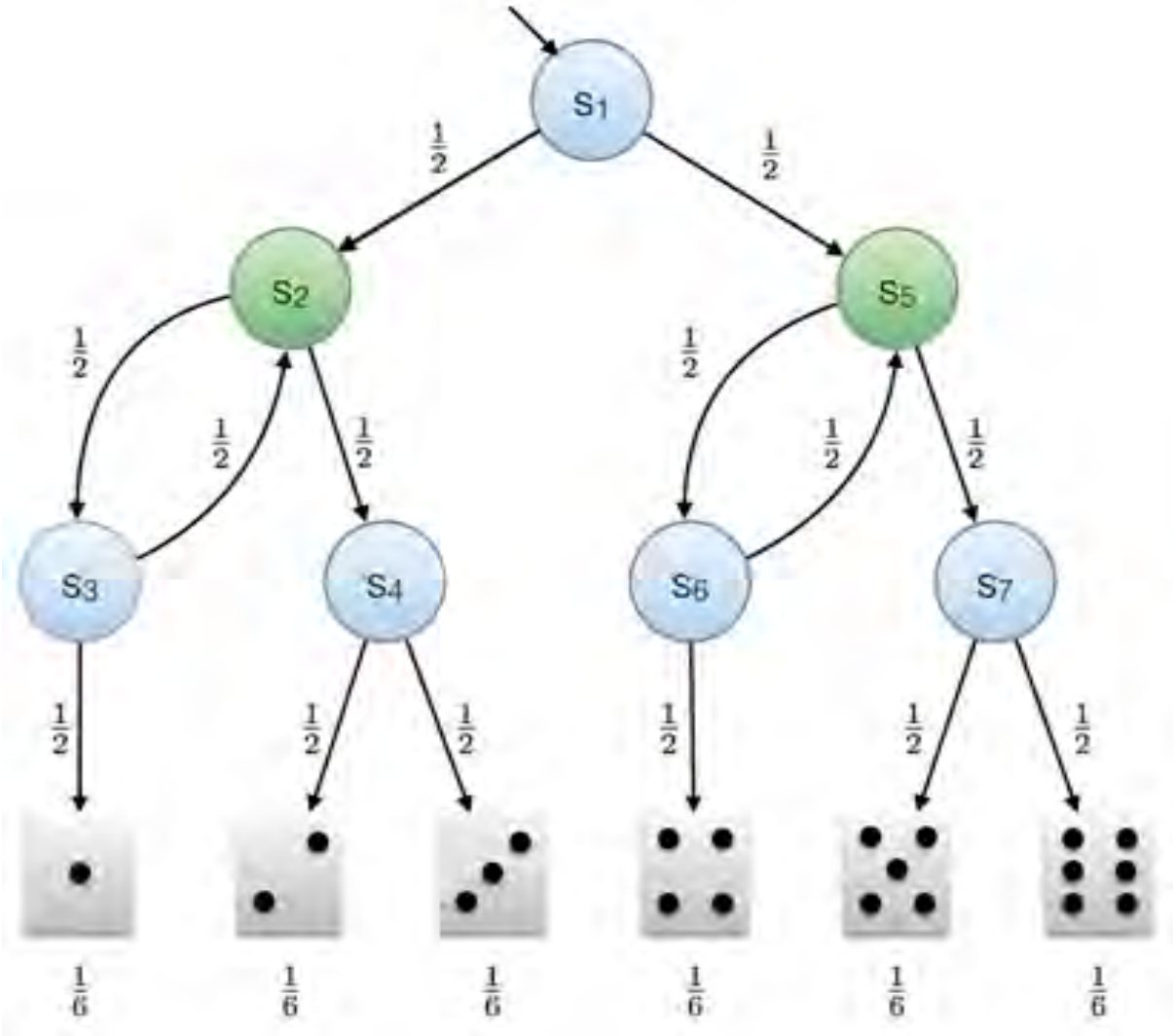
$$x_{s_5} = \frac{1}{2}x_5 + \frac{1}{2}x_4$$

$$x_{s_6} = \frac{1}{2}x_{s_2} + \frac{1}{2}x_6$$

- Gaussian elimination yields:

$$x_{s_5} = \frac{1}{2},\ x_{s_2} = \frac{1}{3},\ x_{s_6} = \frac{1}{6},\ \text{and } \boxed{x_{s_0} = \frac{1}{6}}$$

Correctness



Reachability Probabilities are Pivotal

- Repeated reachability $\Pr(s \models \Box\Diamond G)$:

  Probability to reach a terminal SCC containing a G-state

Reachability Probabilities are Pivotal

- Repeated reachability $\Pr(s \models \Box\Diamond G)$:

  Probability to reach a terminal SCC containing a G-state

- Probabilistic CTL model checking

  Recursive descent on parse tree using reach-probabilities at nodes

Reachability Probabilities are Pivotal

- Repeated reachability $\Pr(s \models \Box\Diamond G)$:

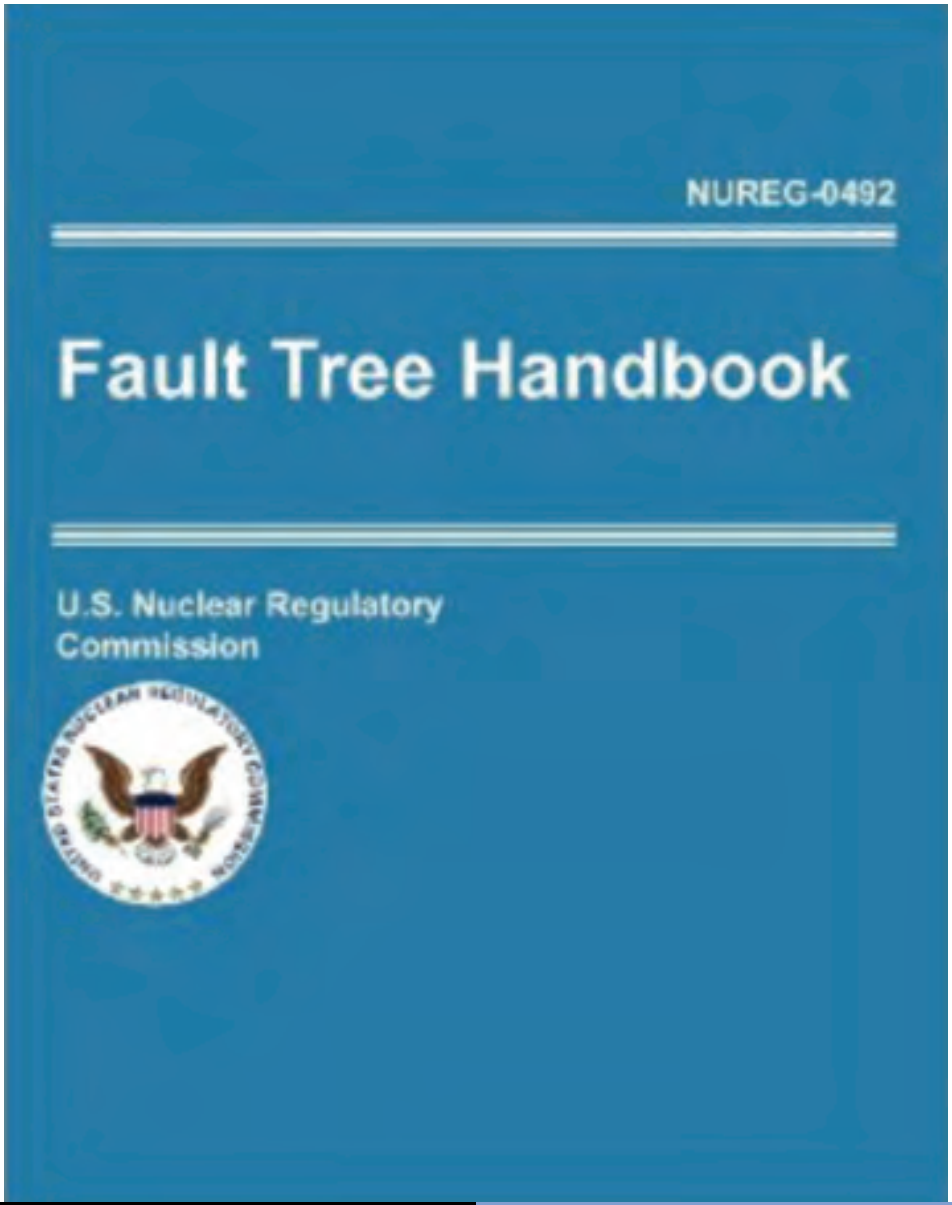
  Probability to reach a terminal SCC containing a G-state

- Probabilistic CTL model checking

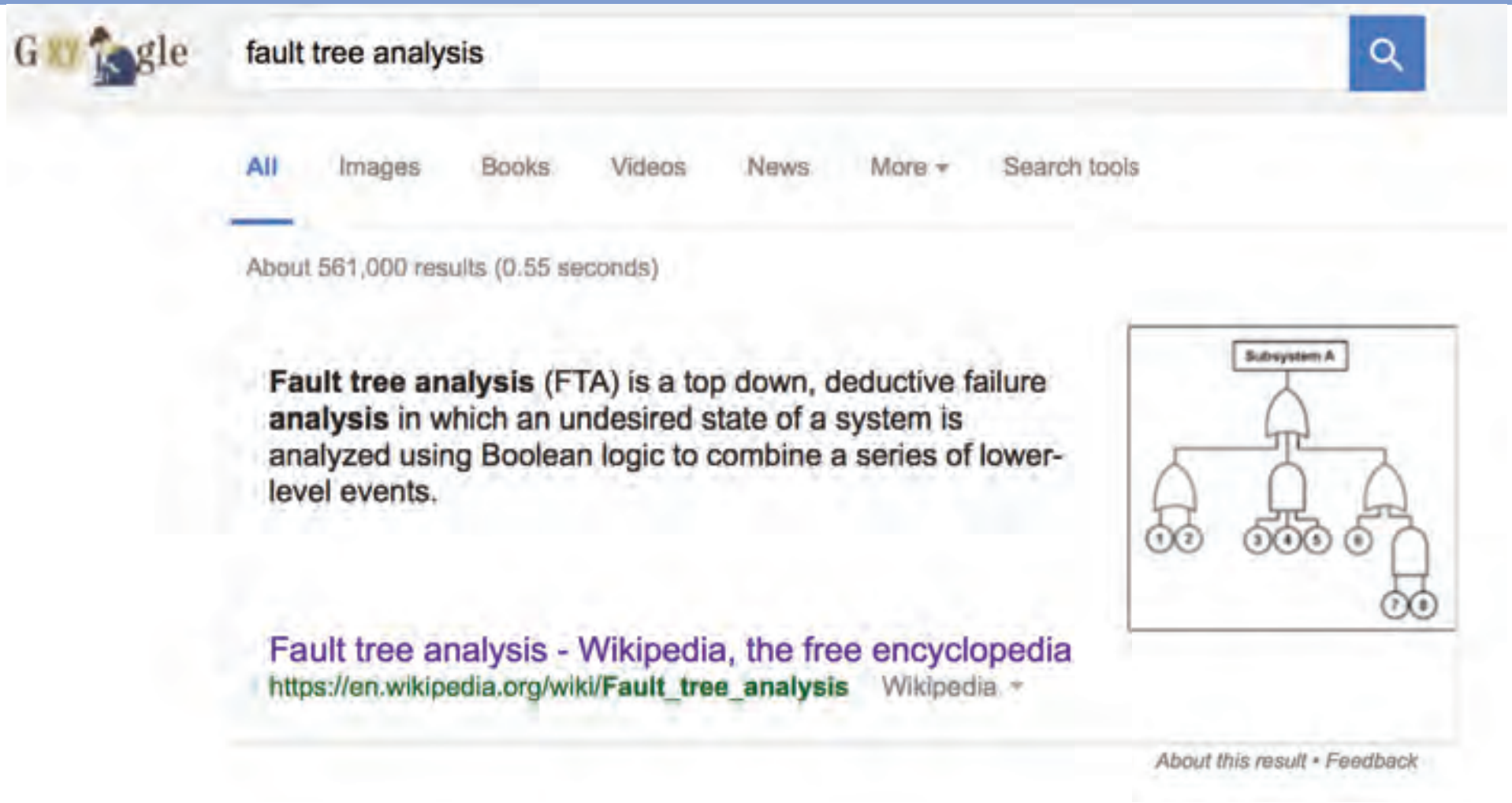
  Recursive descent on parse tree using reach-probabilities at nodes

- LTL formulas $\Pr(s \models \varphi)$:
  1. Transform φ into a deterministic (Rabin) automaton
  2. Take the product of the Markov chain and the automaton
  3. Determine the probability to reach an accepting terminal SCC

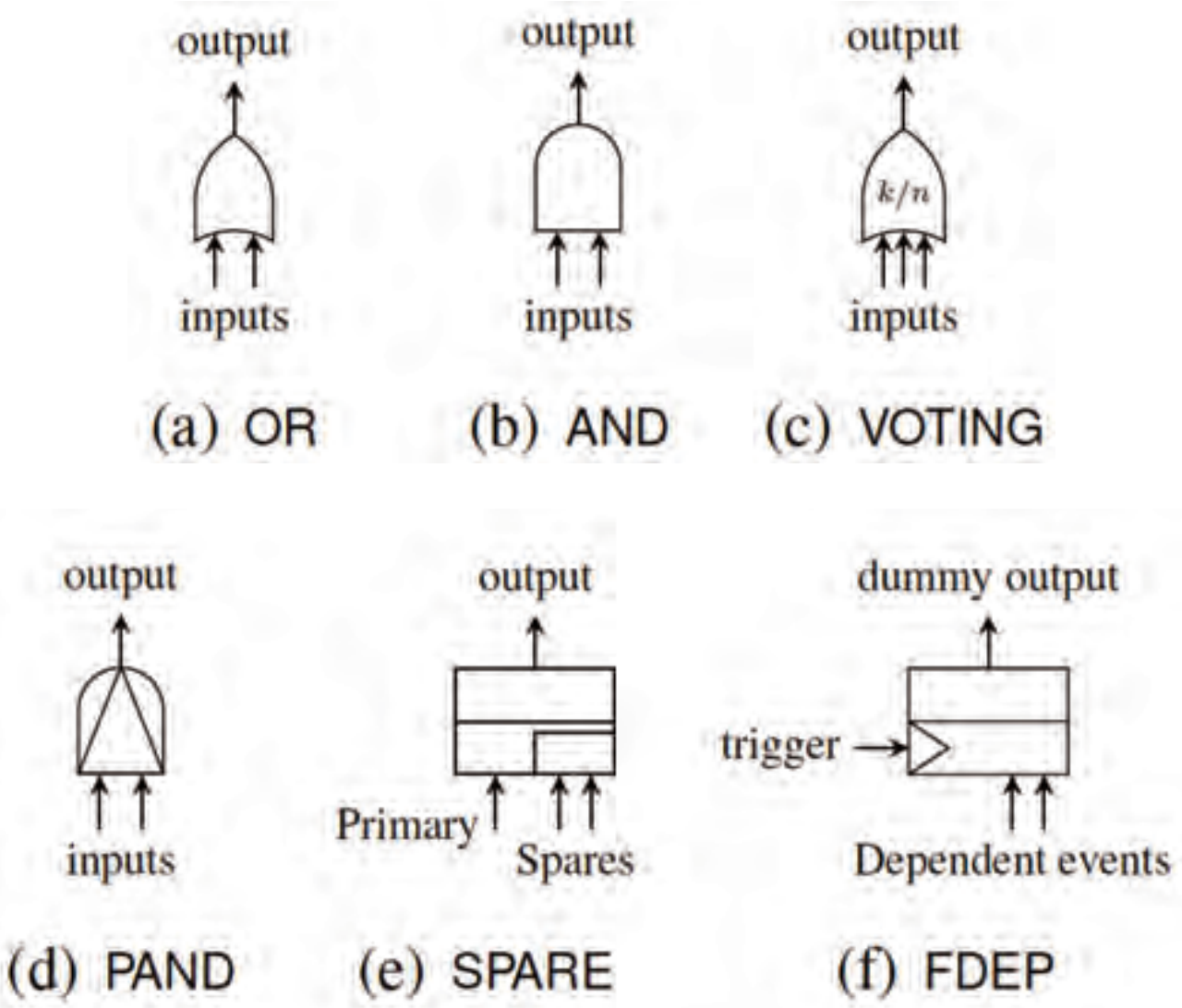
Reliability Engineering



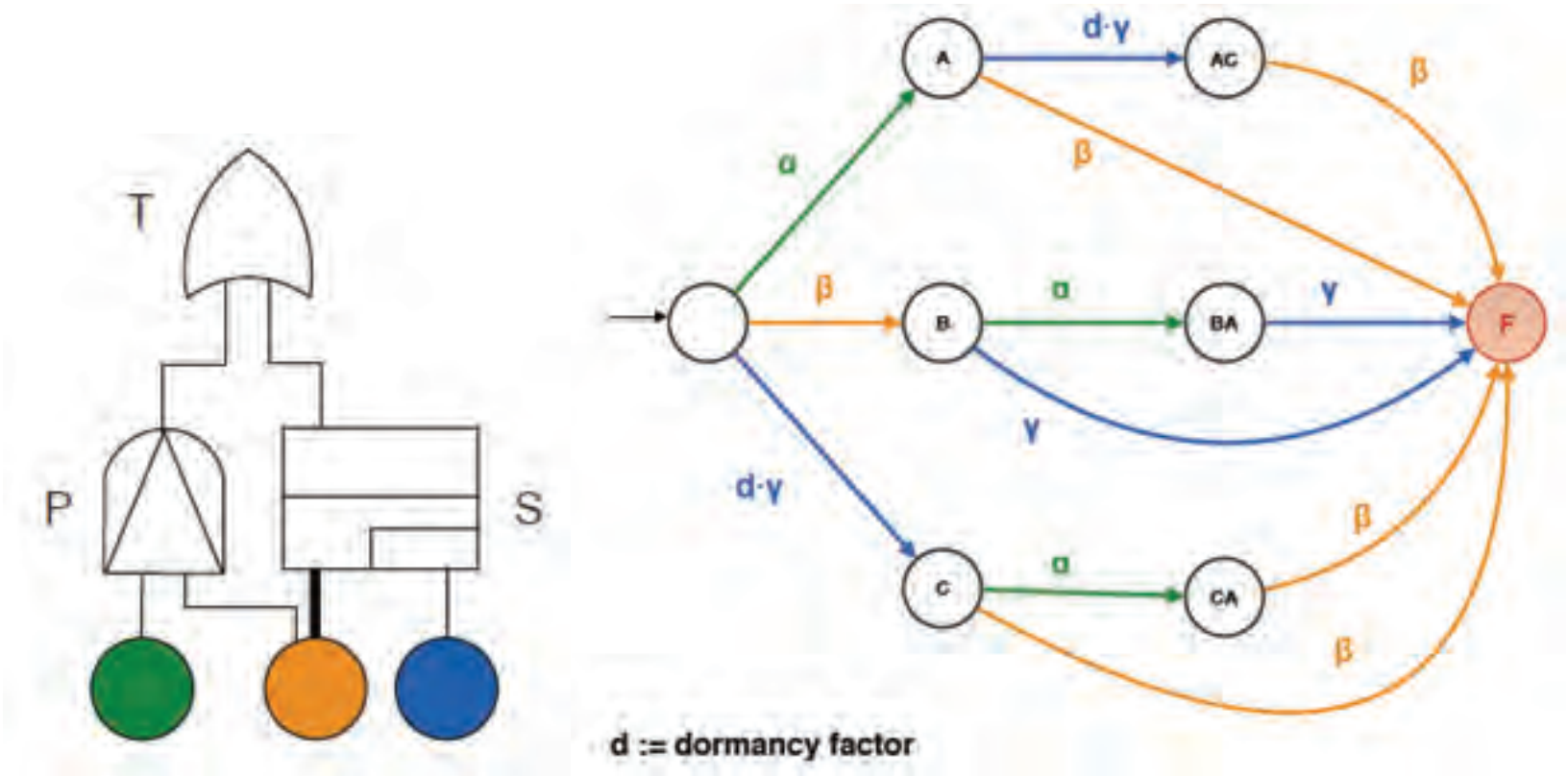
Fault Tree Analysis



Reliability: Dynamic Fault Trees [Dugan *et al.*, 1992]

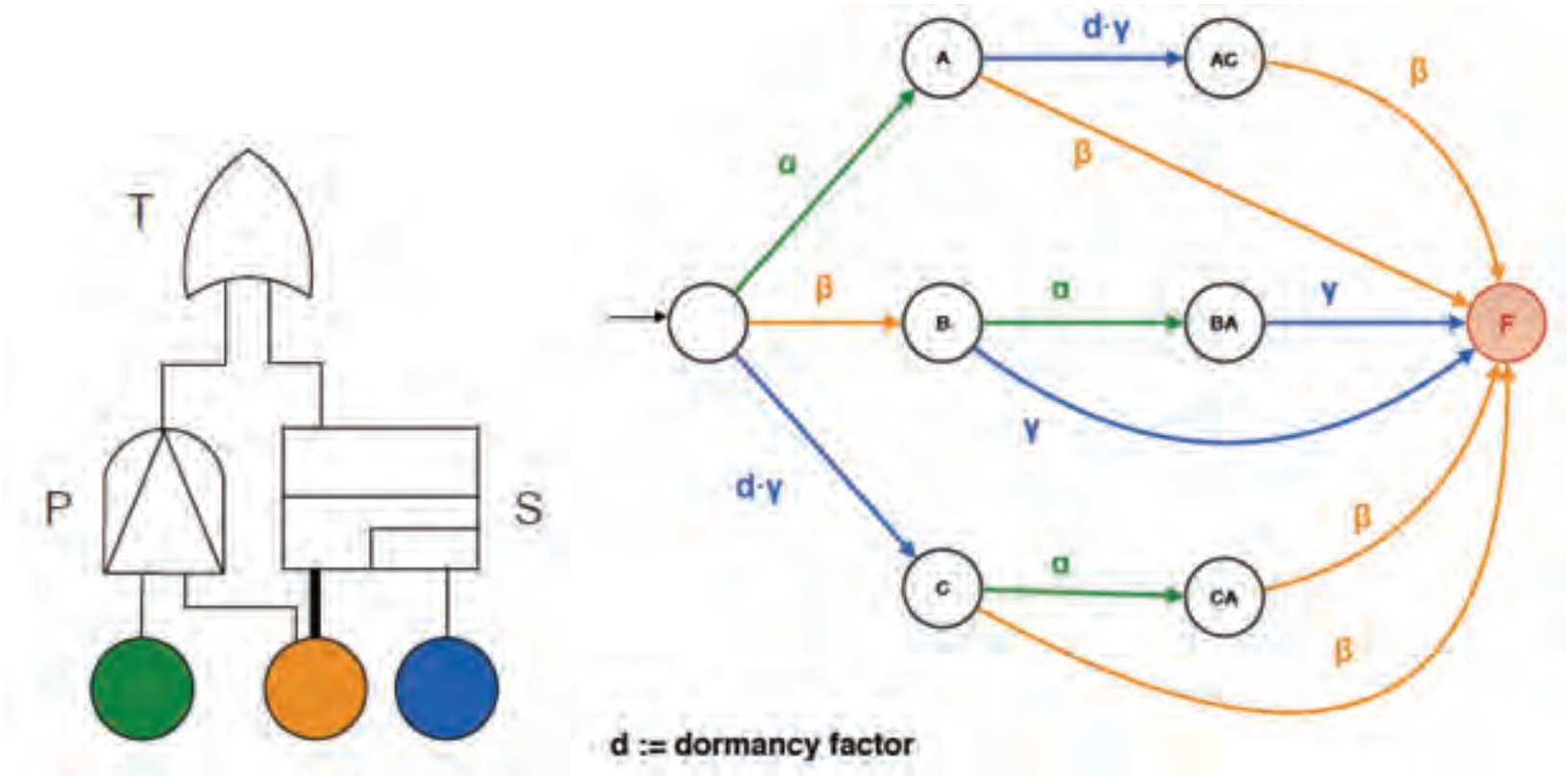


Dynamic Fault Trees



Markov chain process for a DFT

Dynamic Fault Trees



Markov chain process for a DFT

(D)FTs: one of —if not the— most prominent models for risk analysis

Aims: quantify system reliability and availability, MTTF, ...

Probabilistic Model Checkers

- PRISM[1] [Kwiatkowska, Parker *et al.*]
- MRMC [Katoen *et al.*]
- iscasMC [Zhang *et al.*]
- iBioSim [Myers *et al.*]
- GreatSPN [Franceschinis *et al.*]
- SMART [Ciardo *et al.*]
- MarCie [Heiner *et al.*]
- PAT [Song Dong *et al.*]
- storm [Dehnert, Katoen *et al.*]
- .....

Statistical model checkers: Ymer, Vesta, UppAal, APMC, PlasmaLab, ...

[1] Recipient HVC Award 2016.

The Need for Parameter Synthesis

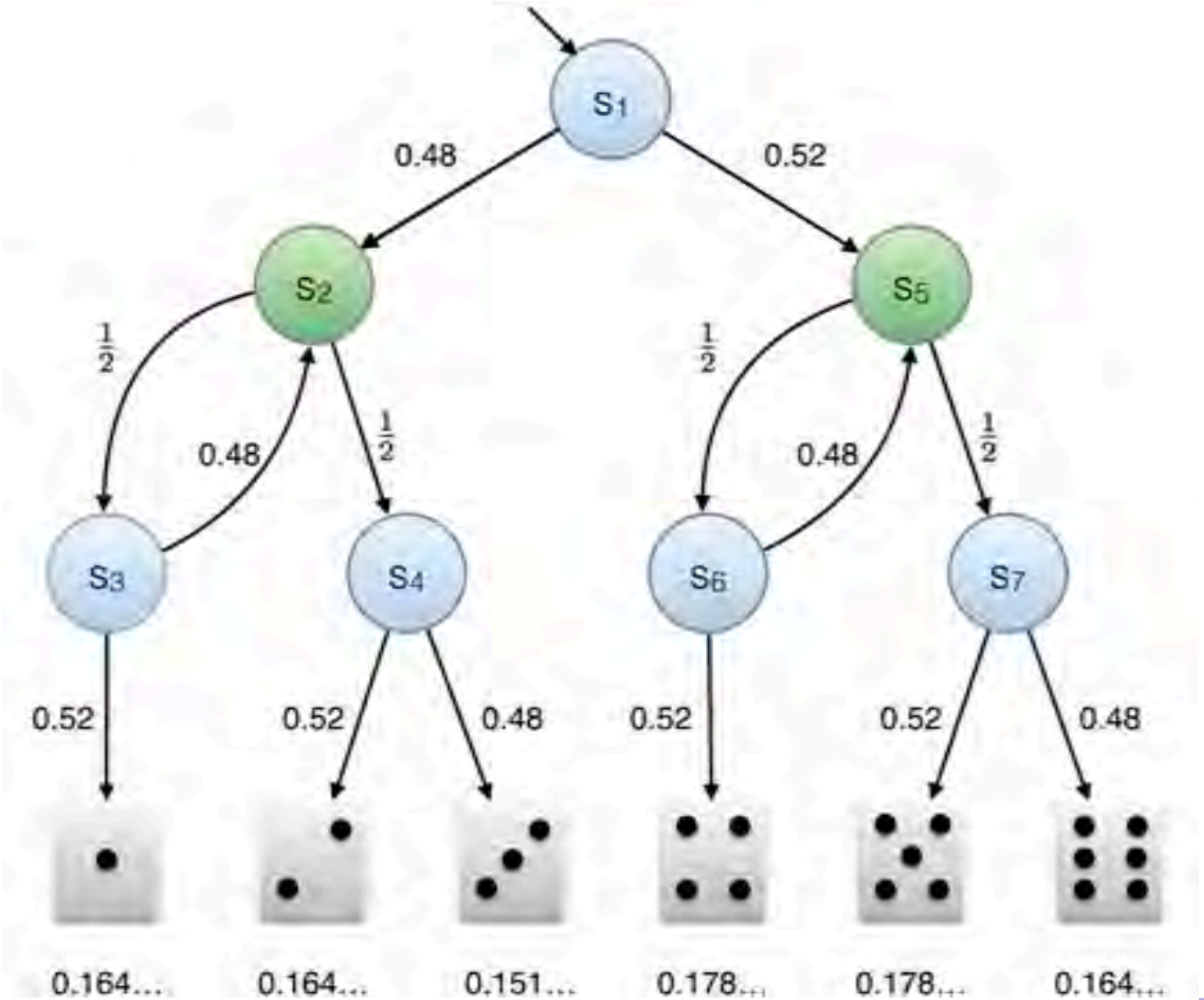
Fact:
Probabilistic model checking is applicable to various areas, e.g.:

- fault trees
- randomised algorithms
- systems biology

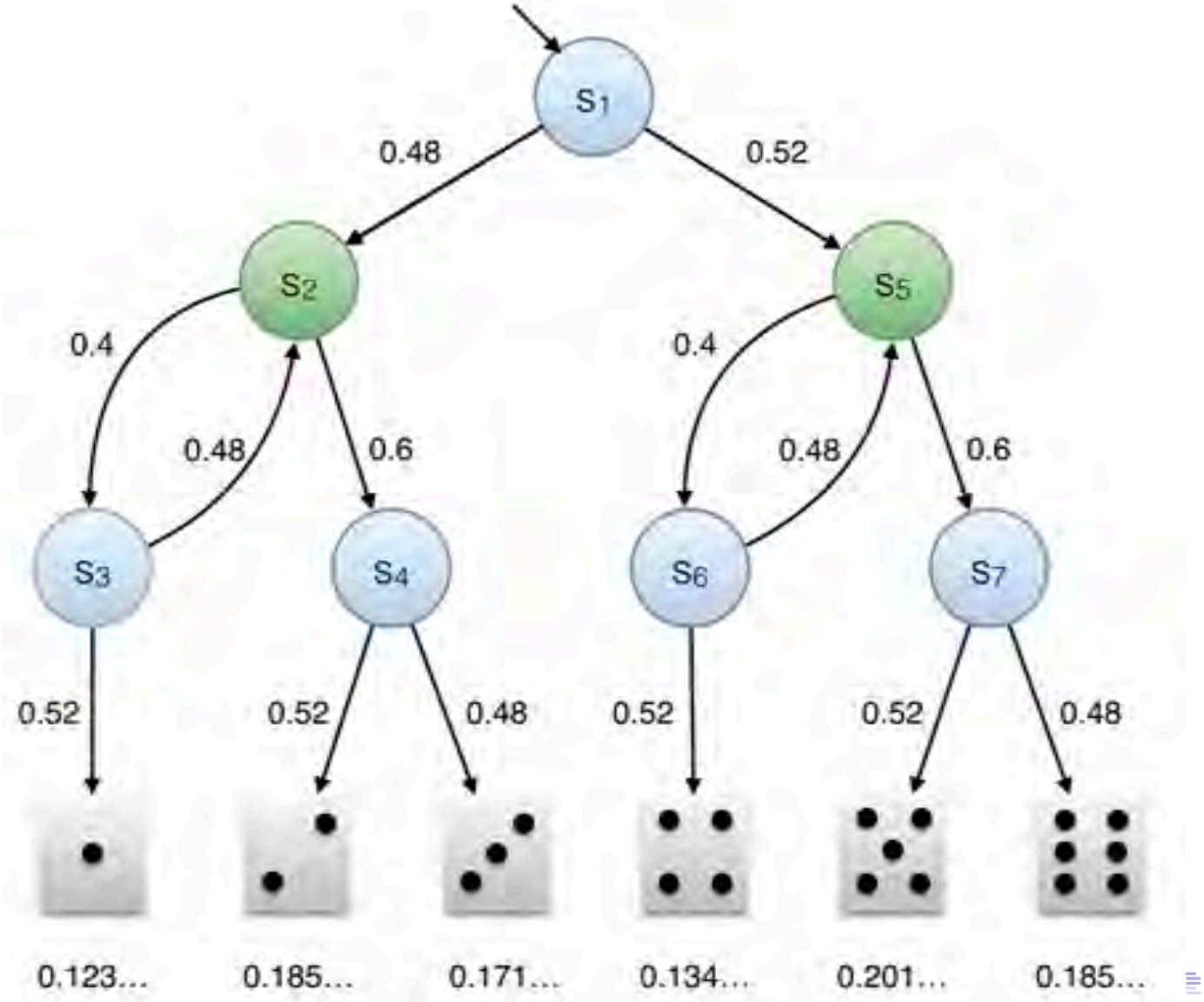
Limitation:
Probabilities need to be known a priori. Is this a valid assumption?
How sensitive are results when transition probabilities fluctuate?

Goal:
Treat parametric models, synthesise "safe" parameter values

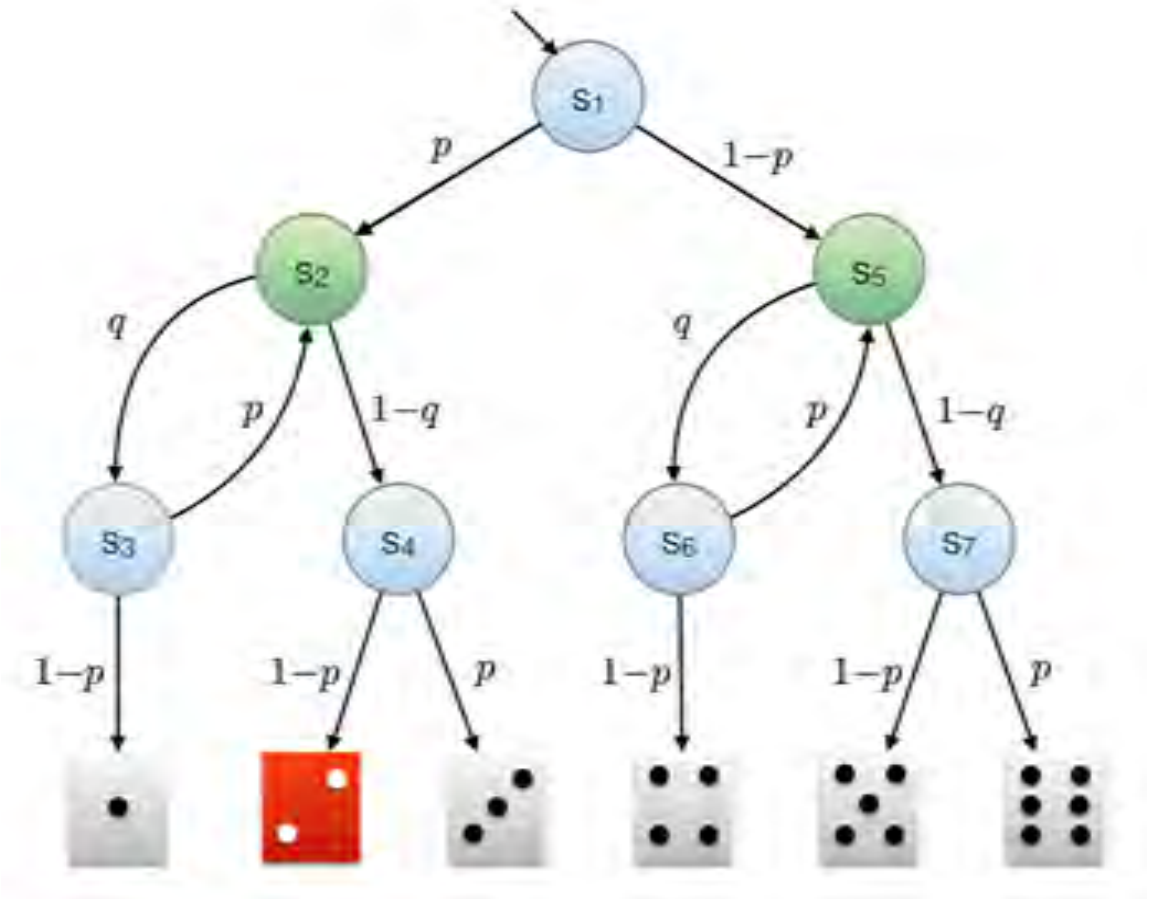
Biased Knuth-Yao's Die



Biased Knuth-Yao's Die

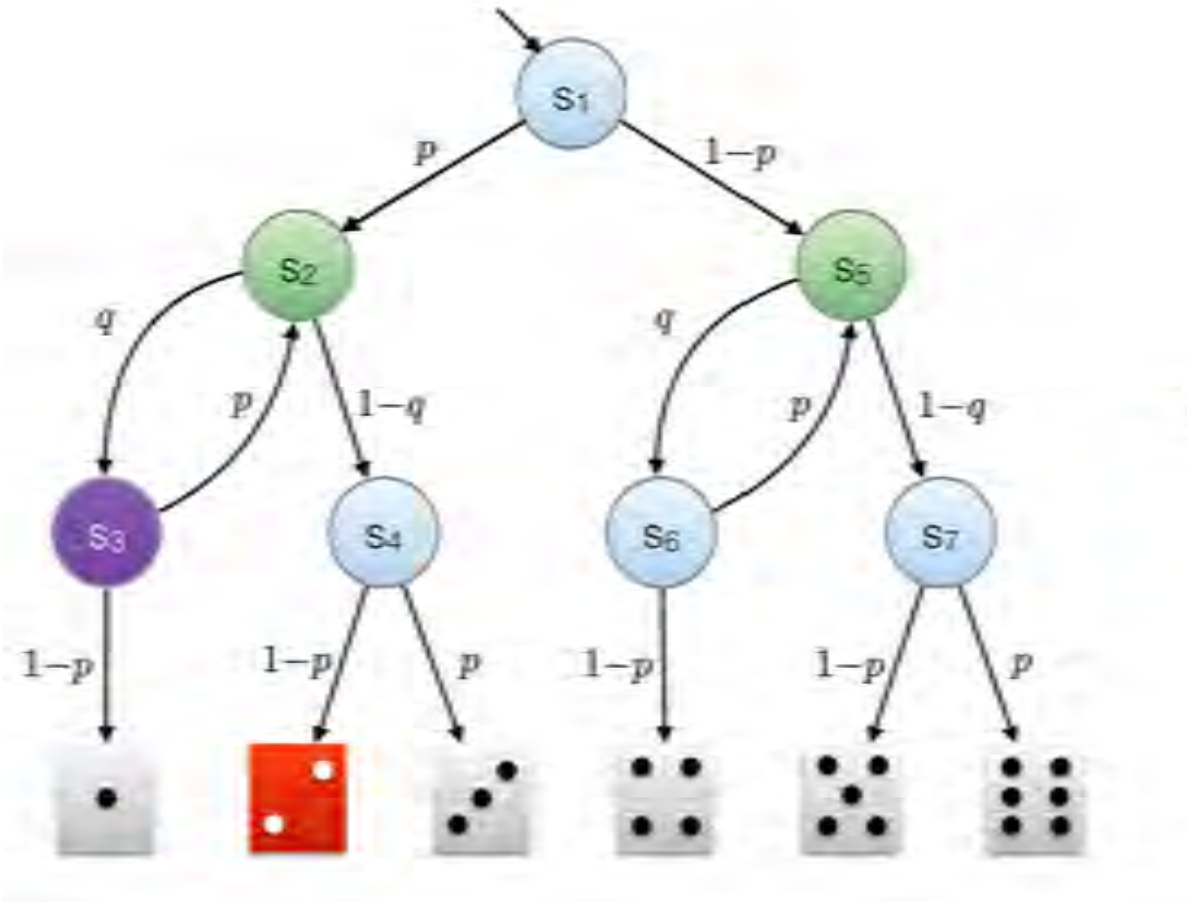


Parametric Knuth-Yao's Die



For which $1/10 \leqslant p \leqslant 9/10$ and $2/5 \leqslant q \leqslant 3/5$ does $\Pr(\Diamond 2) \geqslant 3/20$ hold?

Conditional Probabilities





Parameter Synthesis

Inputs:

1. a (finite) parametric Markov model
2. a property (e.g., reachability, expected reward, conditional reachability)
3. a threshold

Output:

For which parameter values does the pMC satisfy the property with the given threshold?

Parameter Synthesis

Inputs:

1. a (finite) parametric Markov model
2. a property (e.g., reachability, expected reward, conditional reachability)
3. a threshold

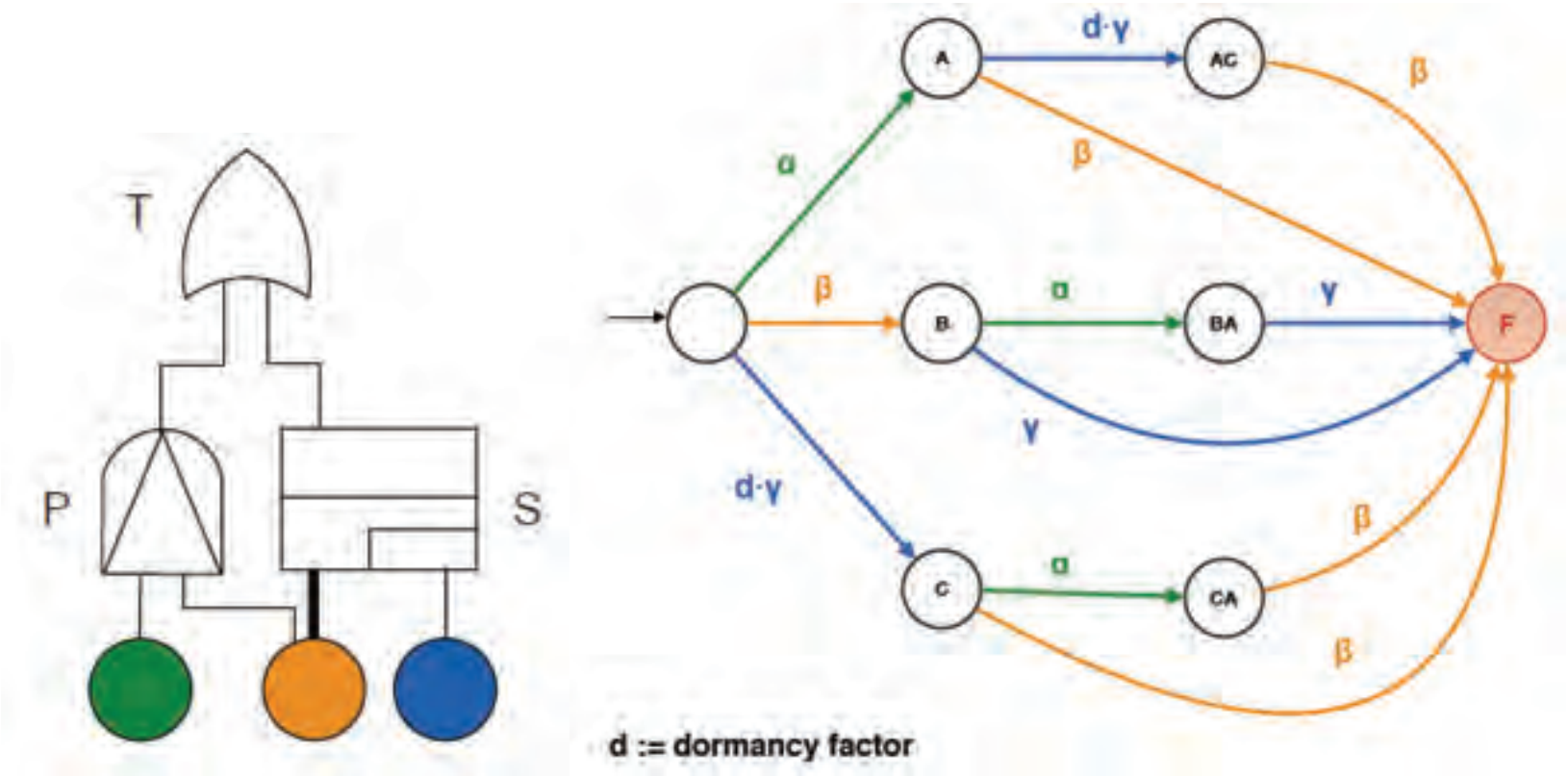
Output:

For which parameter values does the pMC satisfy the property with the given threshold?

Problem instances:

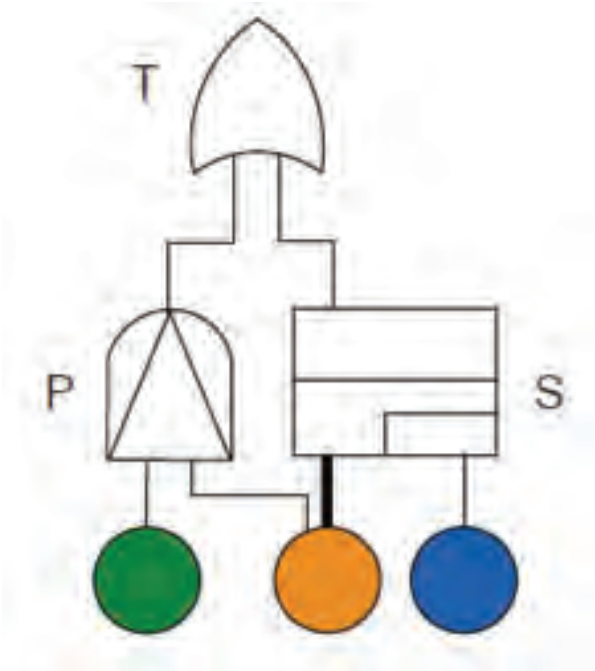
- What is the maximal tolerable message loss ensuring delivery $\geqslant 98\%$?
- ... the tolerable failure rate in a DFT ensuring MTTF $\geqslant 3$ hours?
- .....

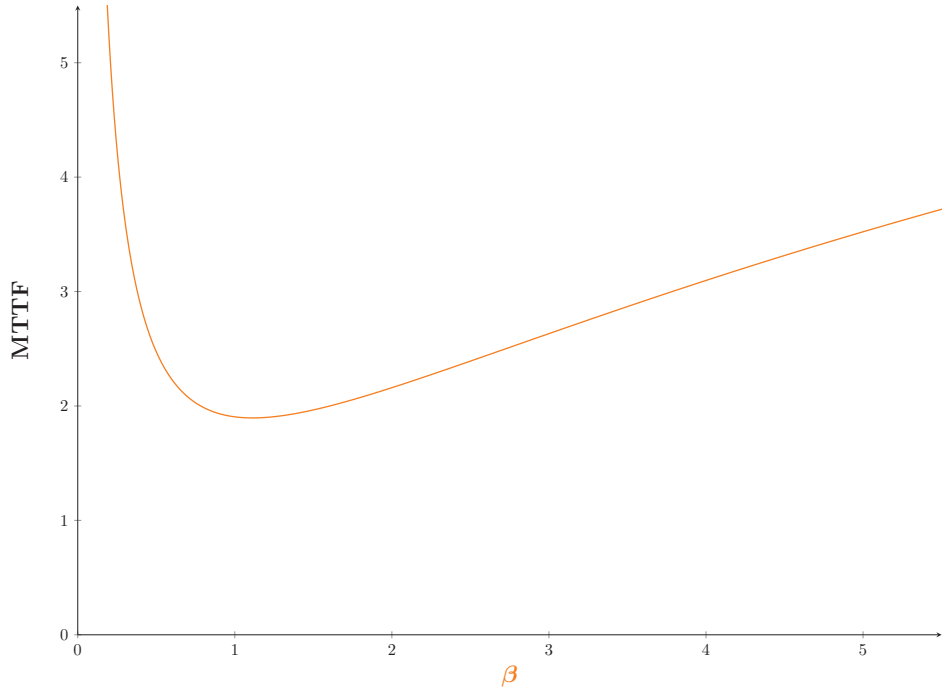
Recall Dynamic Fault Trees



Markov chain process for a DFT

Parametric Fault Trees





Sample parametric DFT and its MTTF

$$\text{MTTF} = \frac{200x^2 + 20x + 201}{x \cdot (20x + 201)} \text{ for } (\alpha, \beta, \gamma, d) = (10, x, 0.1, 0.5)$$

Parameter Synthesis

Aim:

partition the parameter space into safe and unsafe regions

Parameter Synthesis

Aim:

partition the parameter space into safe and unsafe regions

- Region = half-space defined by linear inequalities over the parameters
- A region R for threshold $\leqslant \beta$ is safe if no MC with $v \in R$ exceeds β
- A region R for threshold $\leqslant \beta$ is unsafe if no MC with $v \in R$ is at most β

Parameter Synthesis

Aim:

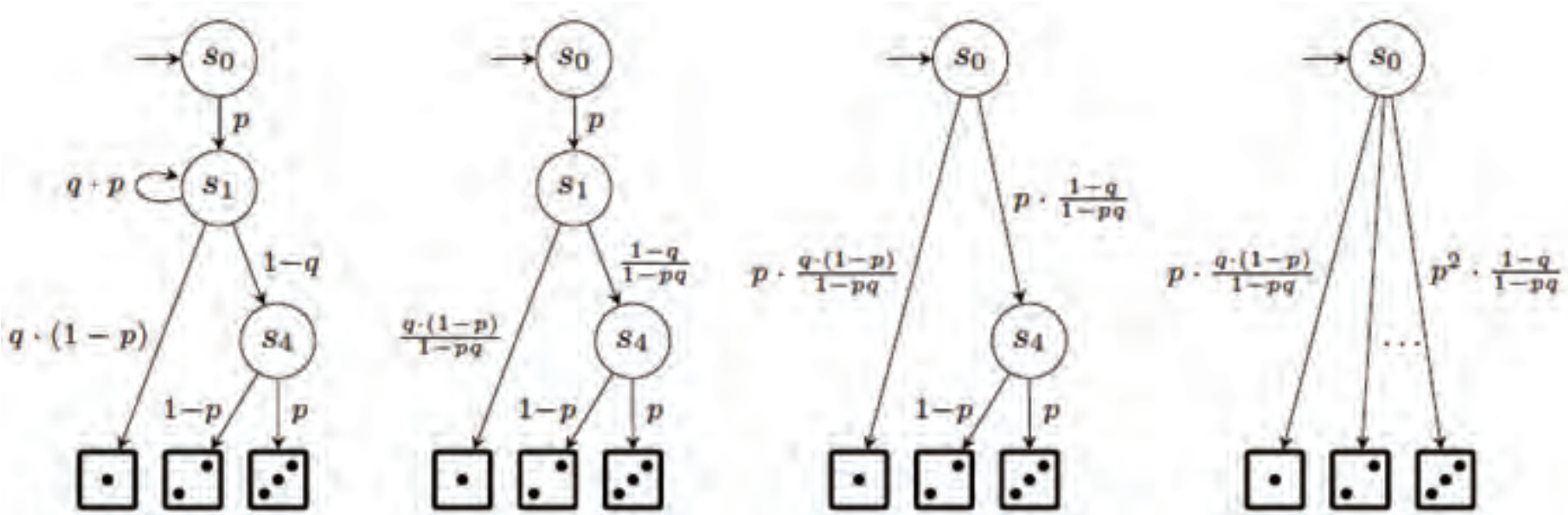
partition the parameter space into safe and unsafe regions

- Region = half-space defined by linear inequalities over the parameters
- A region R for threshold $\leqslant \beta$ is safe if no MC with $v \in R$ exceeds β
- A region R for threshold $\leqslant \beta$ is unsafe if no MC with $v \in R$ is at most β

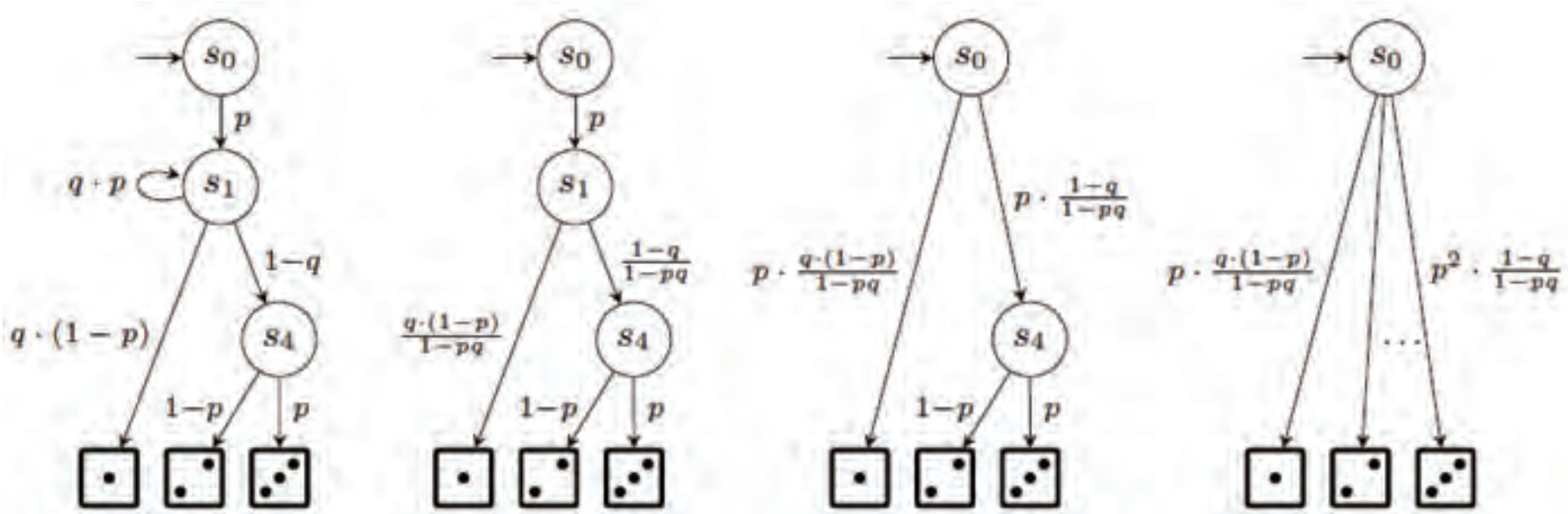
We present two approaches:

1. An exact procedure. How? Using SMT techniques
2. An approximate technique. How? Using parameter lifting

Computing Rational Functions [Daws, 2004]

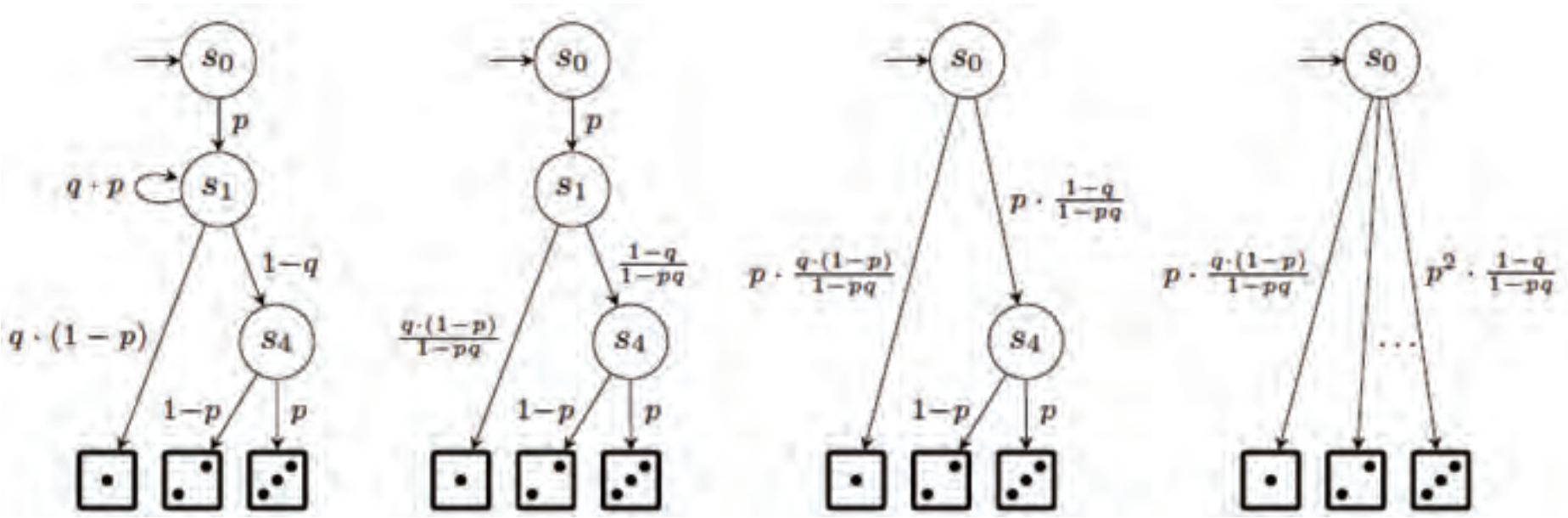


Computing Rational Functions [Daws, 2004]



$$\Pr(s_0 \models \Diamond(1 \text{ or } 3)) \leqslant 1/3 \quad \text{iff} \quad p \cdot q \cdot \frac{1-p}{1-p \cdot q} + p^2 \cdot \frac{1-q}{1-p \cdot q} \leqslant 1/3$$

Computing Rational Functions [Daws, 2004]



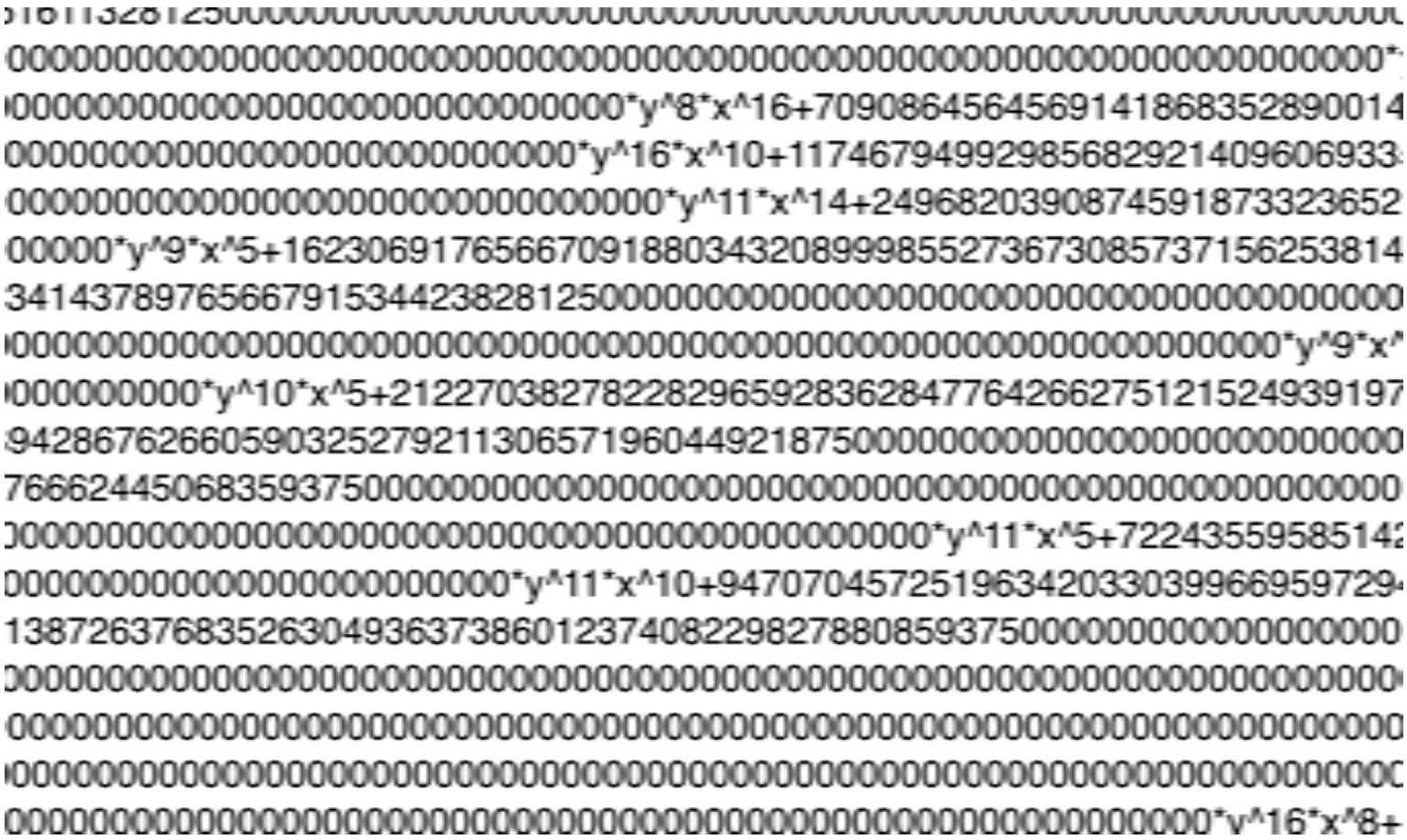
$$\Pr(s_0 \models \Diamond(1 \text{ or } 3)) \leqslant 1/3 \quad \text{iff} \quad p \cdot q \cdot \frac{1-p}{1-p \cdot q} + p^2 \cdot \frac{1-q}{1-p \cdot q} \leqslant 1/3$$

This may yield large high-degree rational functions.

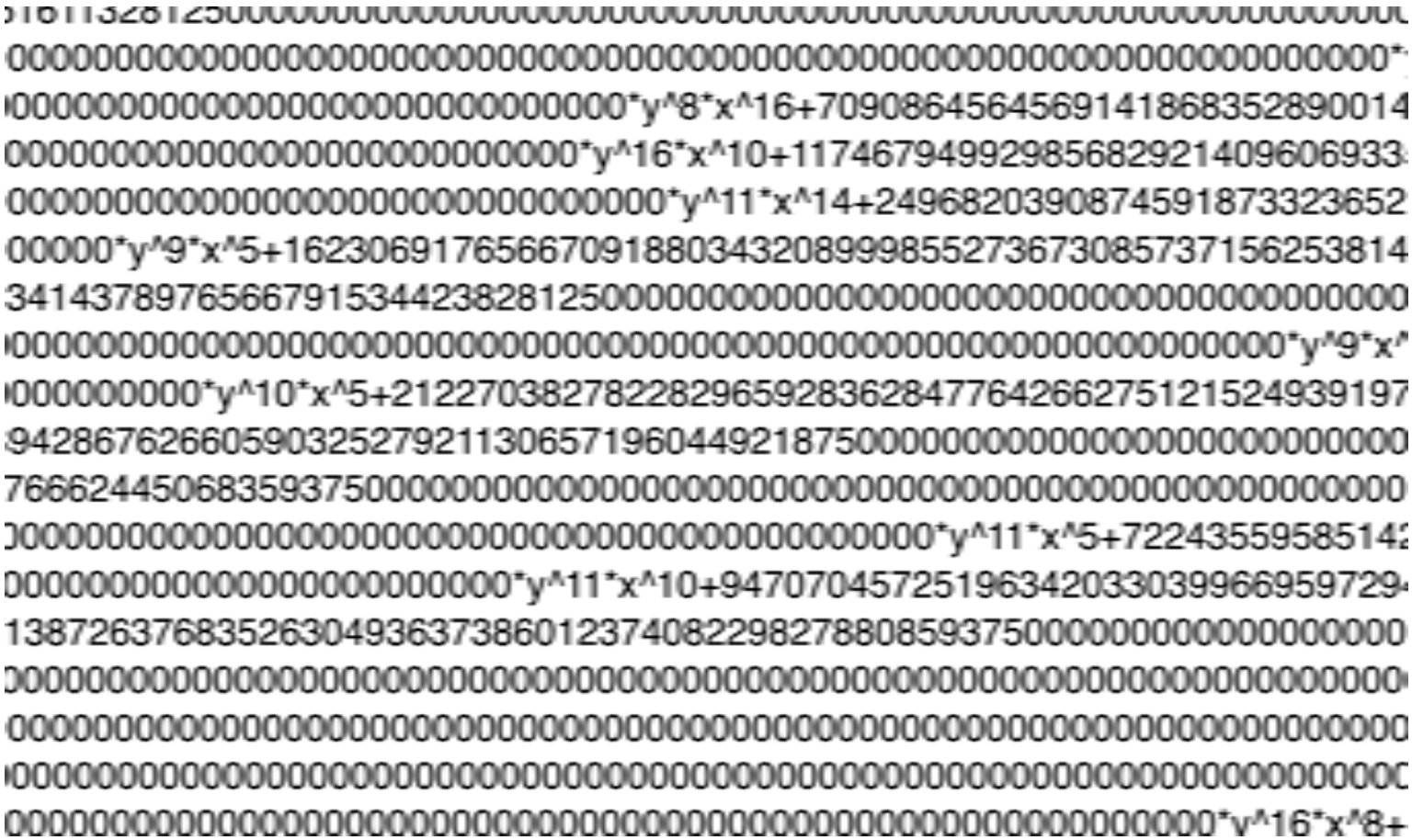
Resulting Rational Functions



Zooming In


0000000000000000000000000000000000000000000000000000000000000000*
0000000000000000000000000000*y^8*x^16+709086456456914186835289001
0000000000000000000000000000*y^16*x^10+11746794992985682921409606933
0000000000000000000000000000*y^11*x^14+24968203908745918733236 52
00000*y^9*x^5+162306917656670918803432089998552736730857371562538 14
3414378976566791534423828125000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000*y^9*x^
0000000000*y^10*x^5+212270382782282965928362847764266275121524939197
9428676266059032527921130657196044921875000000000000000000000000
7666244506835937500000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000*y^11*x^5+722435958514
0000000000000000000000*y^11*x^10+947070457251963420330399669597294
1387263768352630493637386012374082298278808593750000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000*y^16*x^8+

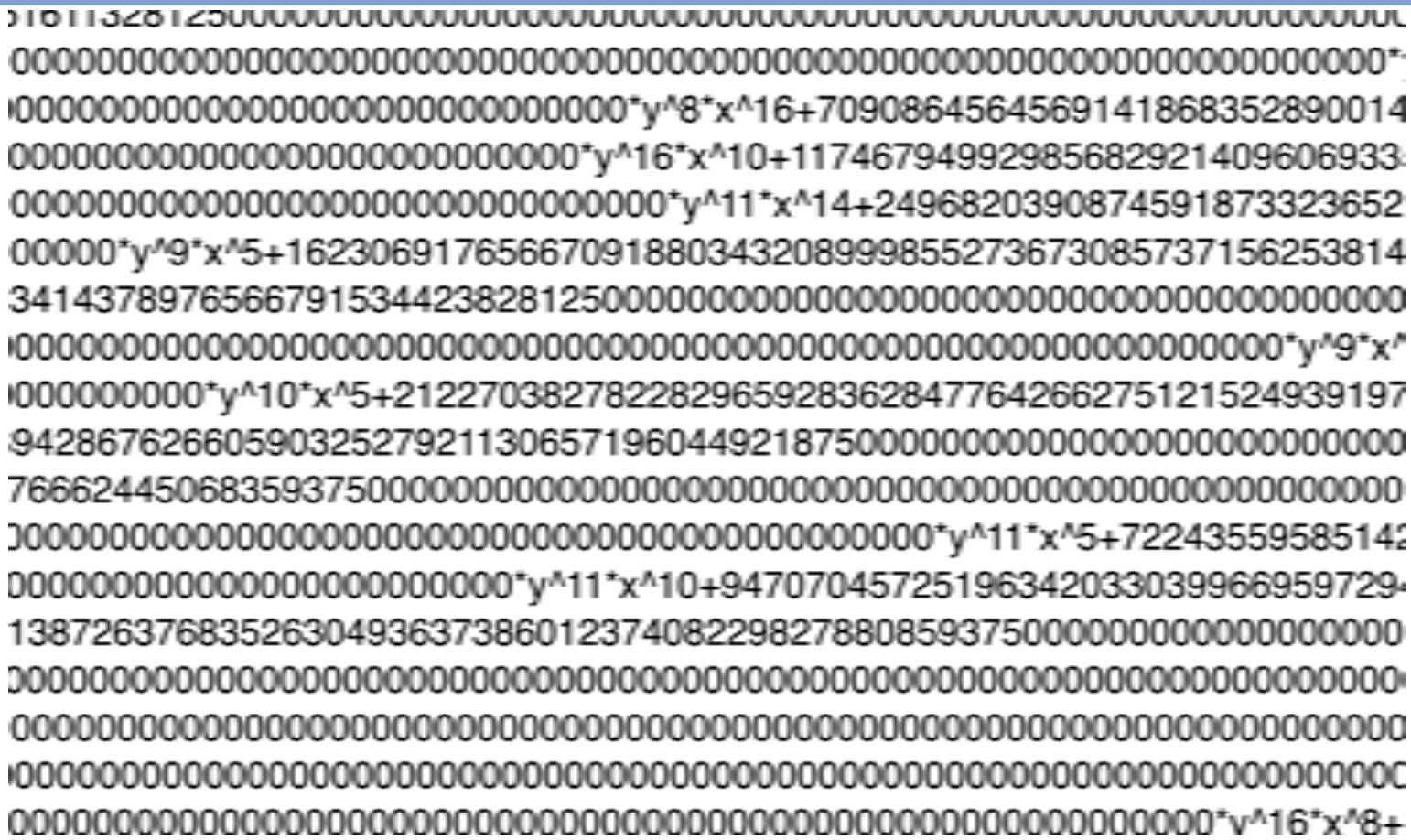
Zooming In



41 states, 138 transitions, 2 parameters:

numerator = 48 terms, denominator = product of 48 (linear) polynomials

Zooming In

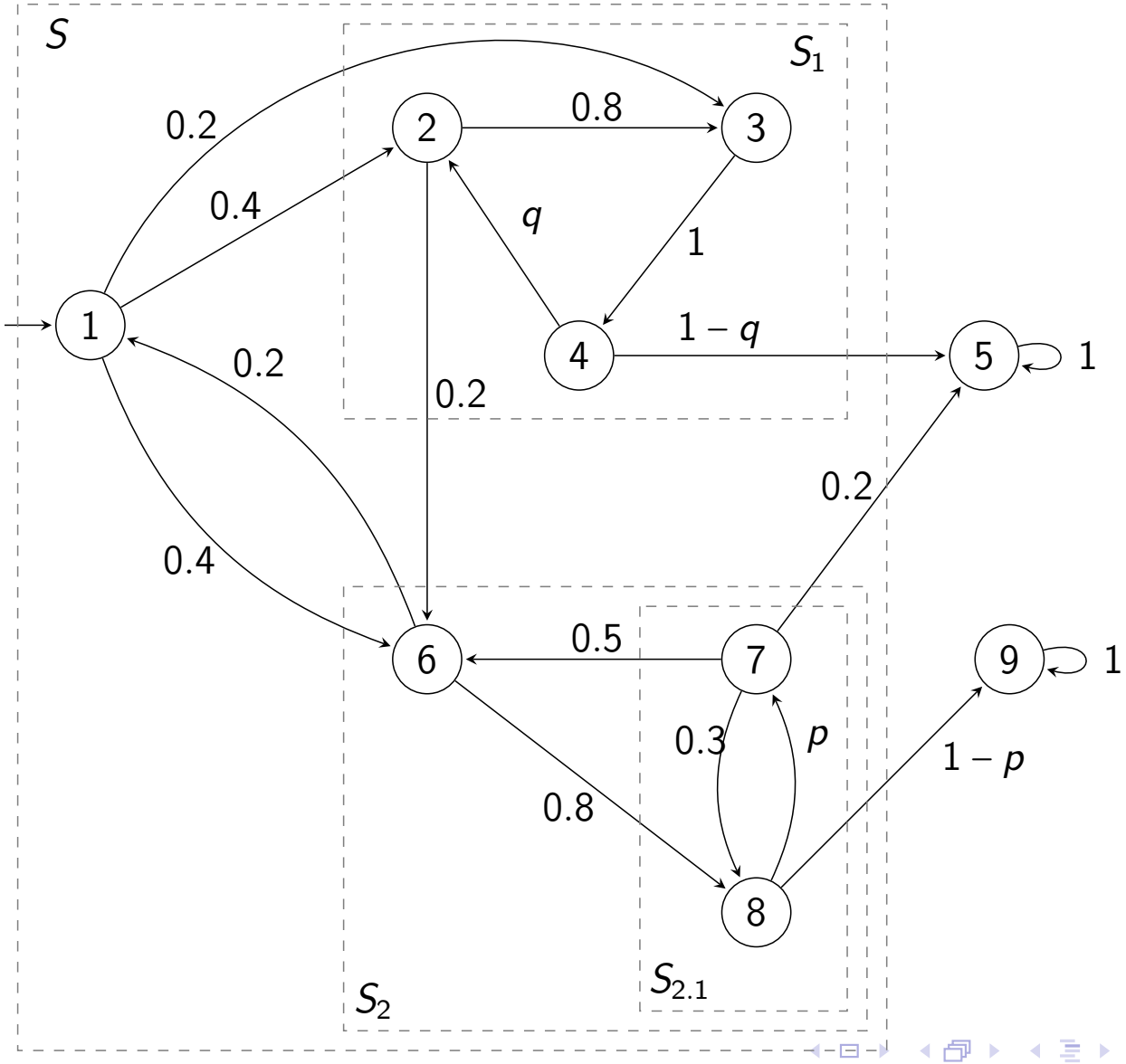


41 states, 138 transitions, 2 parameters:

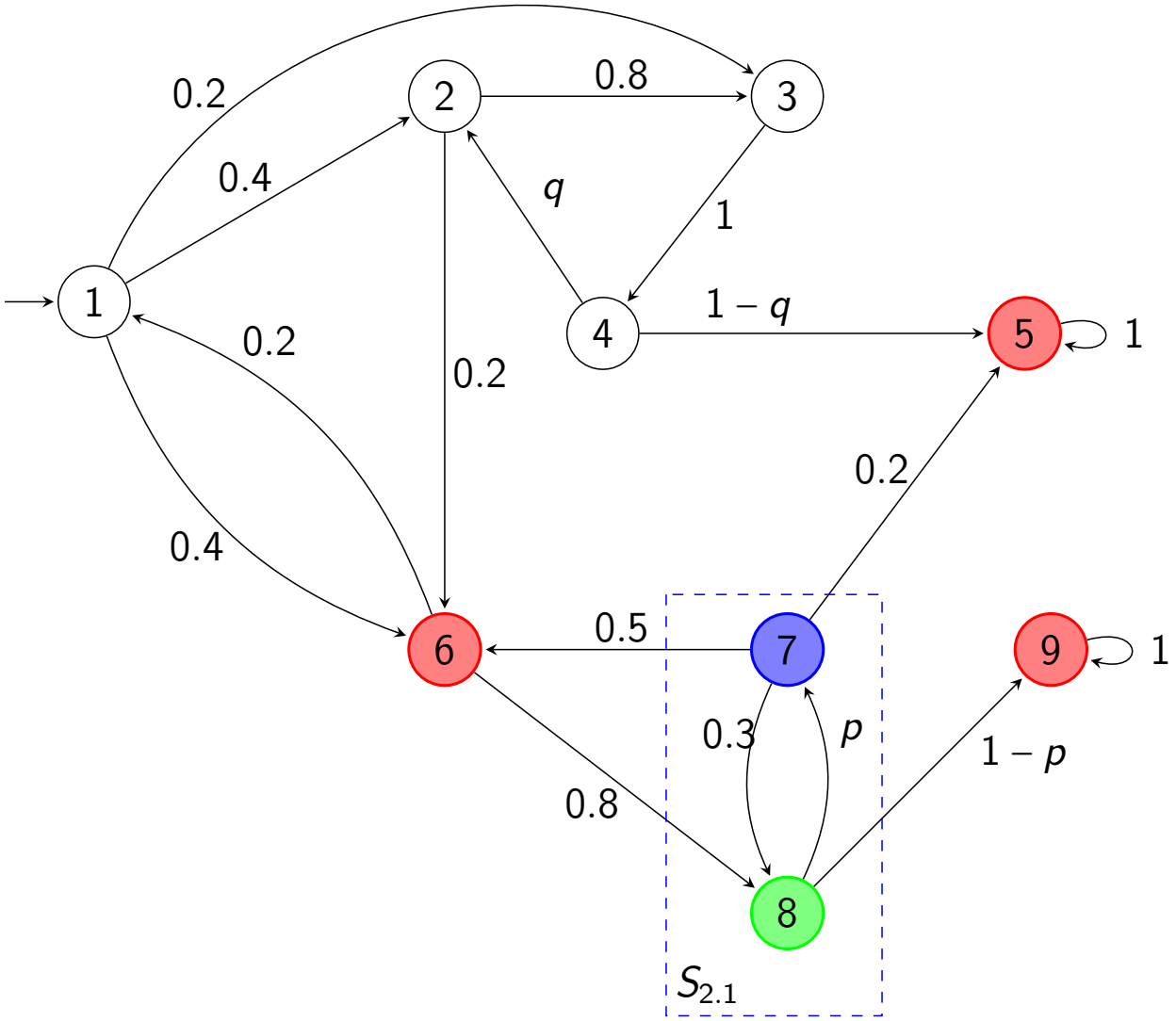
numerator = 48 terms, denominator = product of 48 (linear) polynomials

$\Rightarrow$ Use bisimulation, SCC-decomposition and efficient gcd-computation

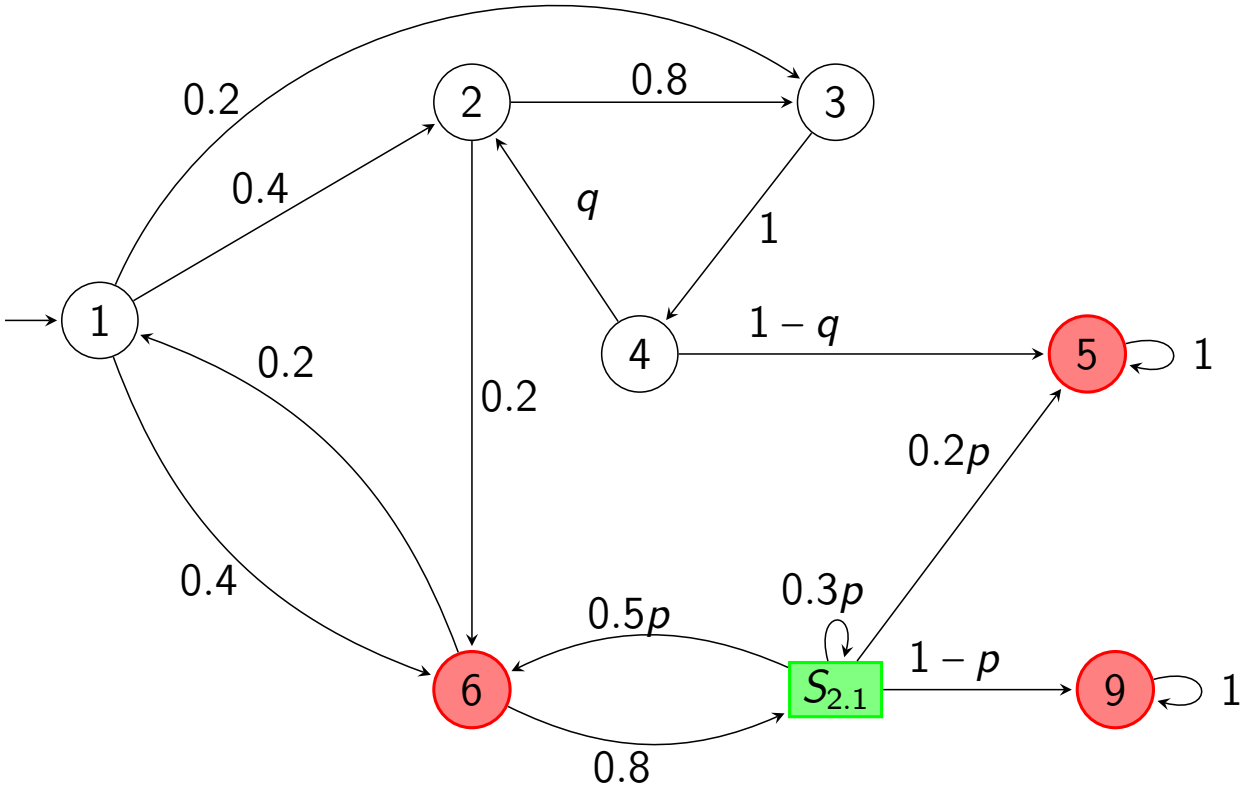
Hierarchical SCC Decomposition [Jansen *et al.*, 2014]



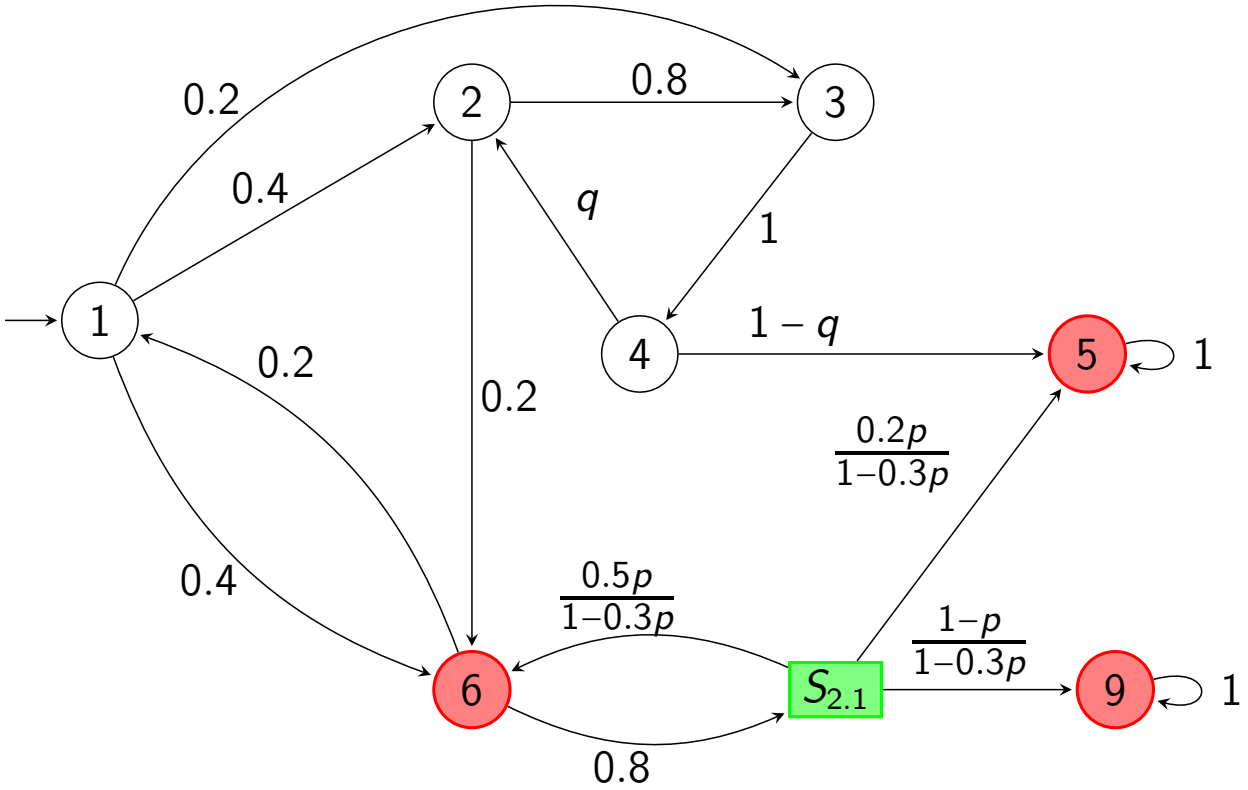
Hierarchical SCC Decomposition [Jansen *et al.*, 2014]



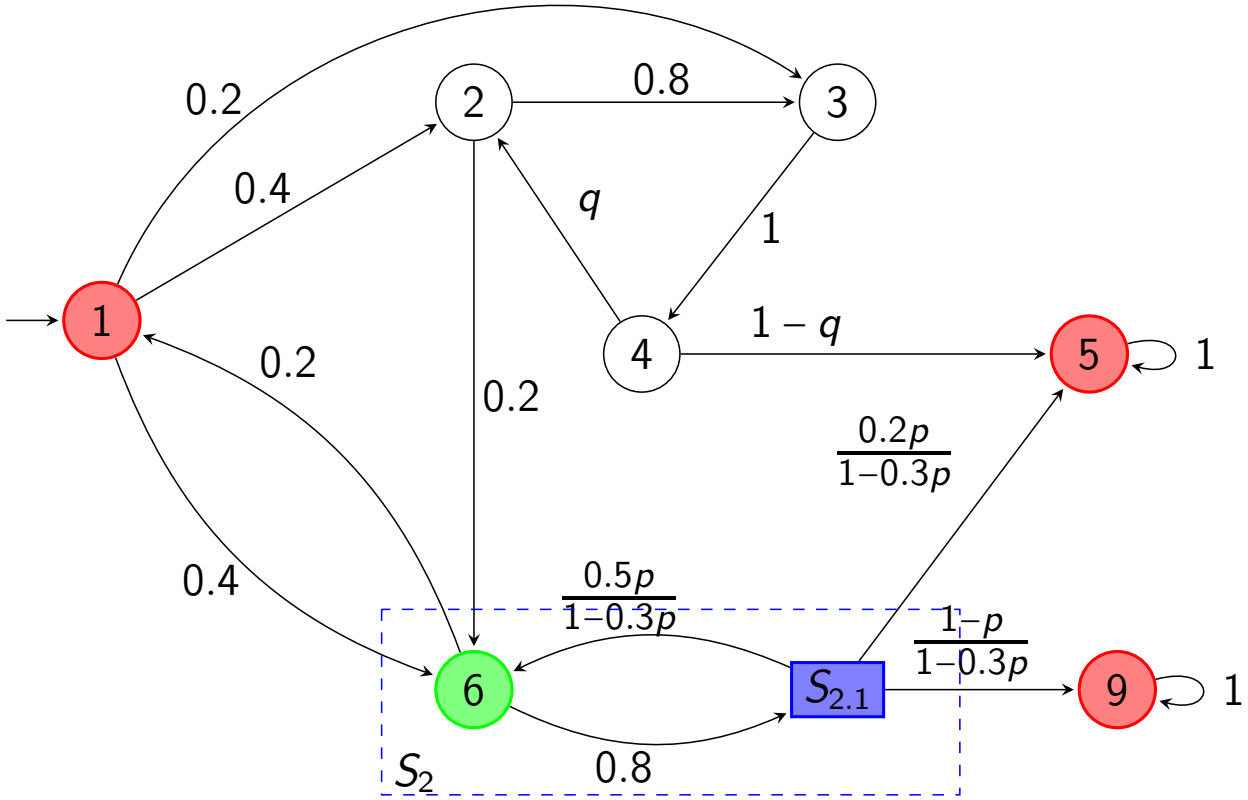
Hierarchical SCC Decomposition [Jansen *et al.*, 2014]



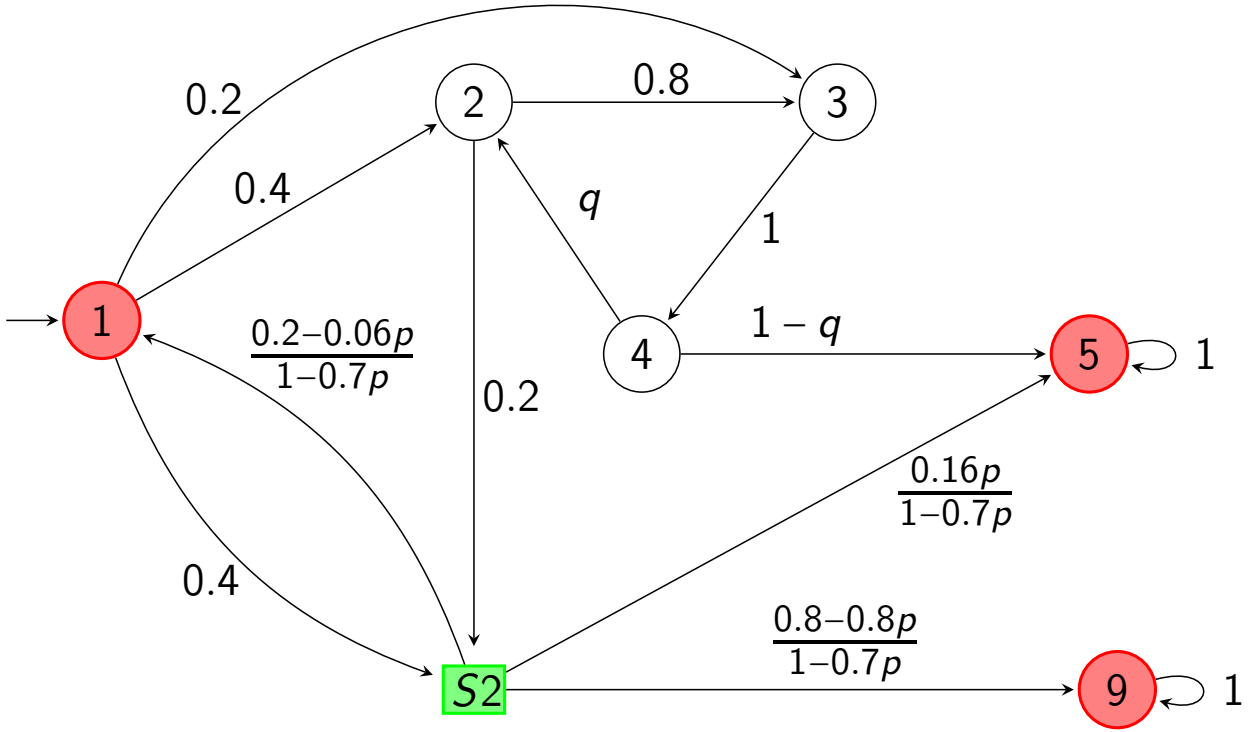
Hierarchical SCC Decomposition [Jansen *et al.*, 2014]



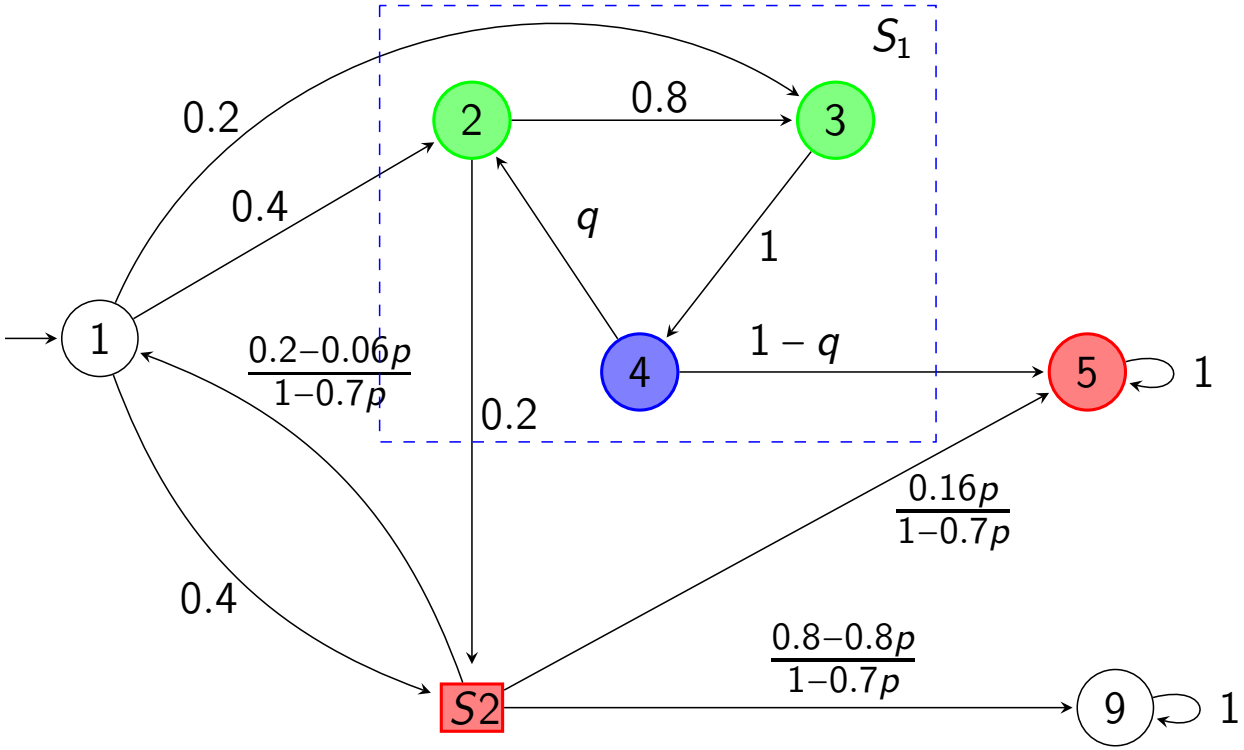
Hierarchical SCC Decomposition [Jansen *et al.*, 2014]



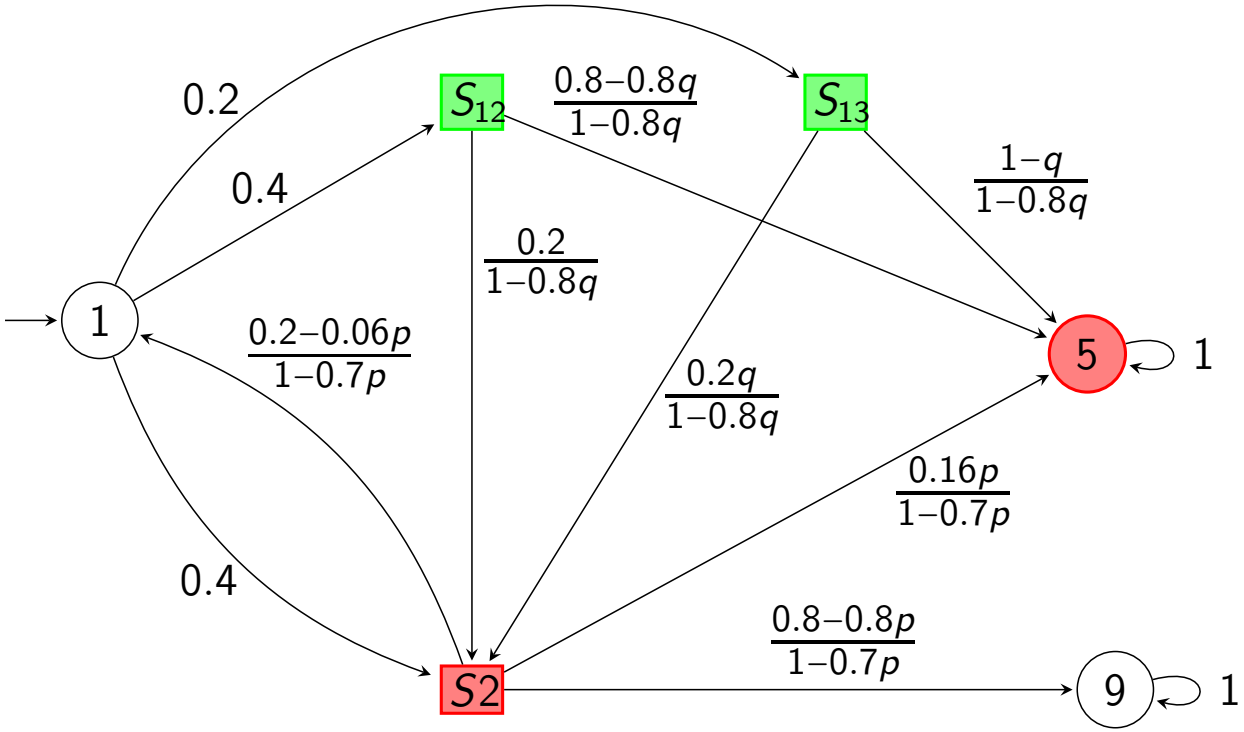
Hierarchical SCC Decomposition [Jansen *et al.*, 2014]



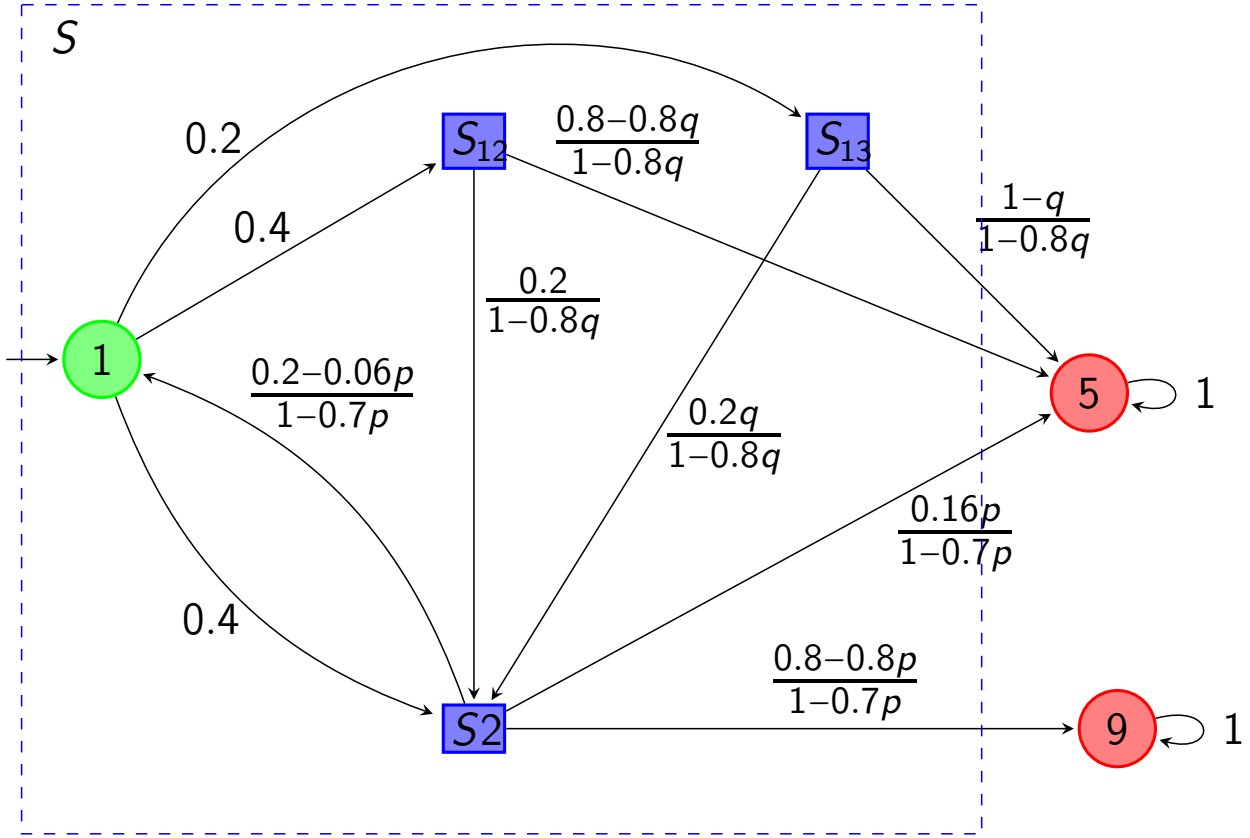
Hierarchical SCC Decomposition [Jansen *et al.*, 2014]



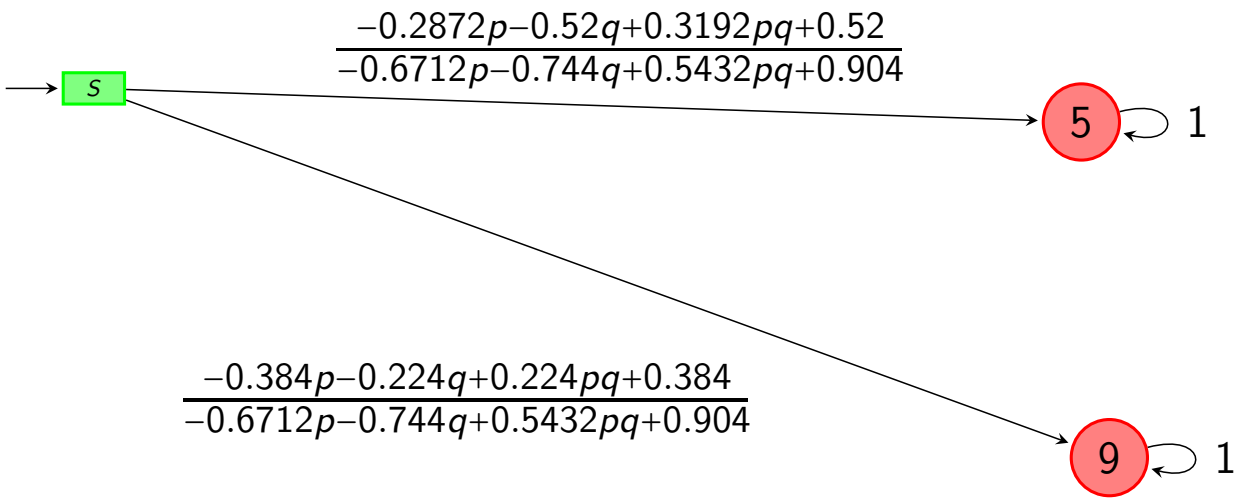
Hierarchical SCC Decomposition [Jansen *et al.*, 2014]



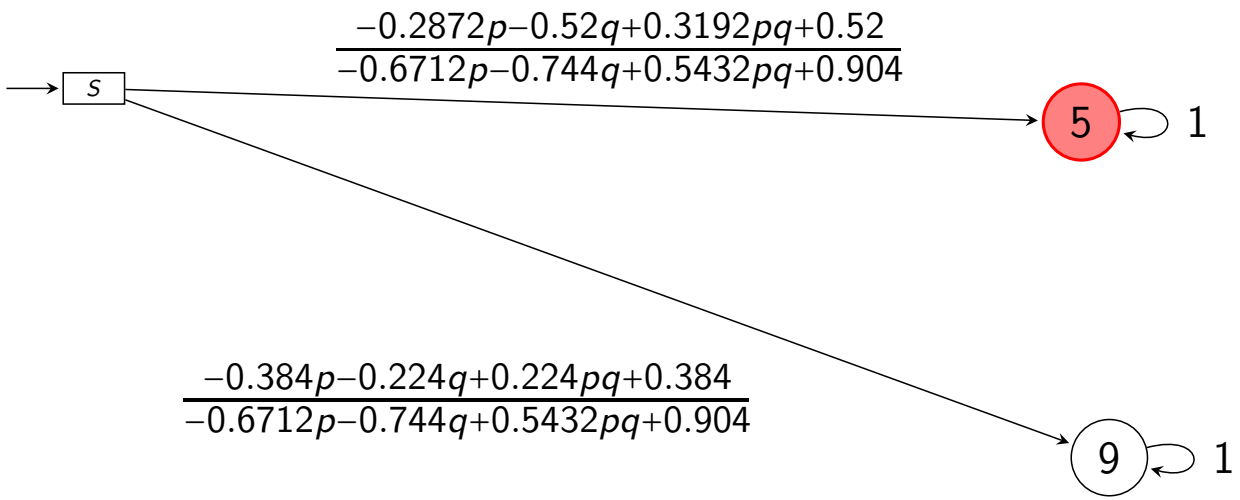
Hierarchical SCC Decomposition [Jansen *et al.*, 2014]



Hierarchical SCC Decomposition [Jansen *et al.*, 2014]



Hierarchical SCC Decomposition [Jansen *et al.*, 2014]



For which (combinations of) values for p and q is the probability of reaching 5 smaller than $c \in [0,1]$?
$\Rightarrow$ Evaluate rational function.

Exploiting SMT

Goal: partition parameter space in regions R that are either safe or unsafe

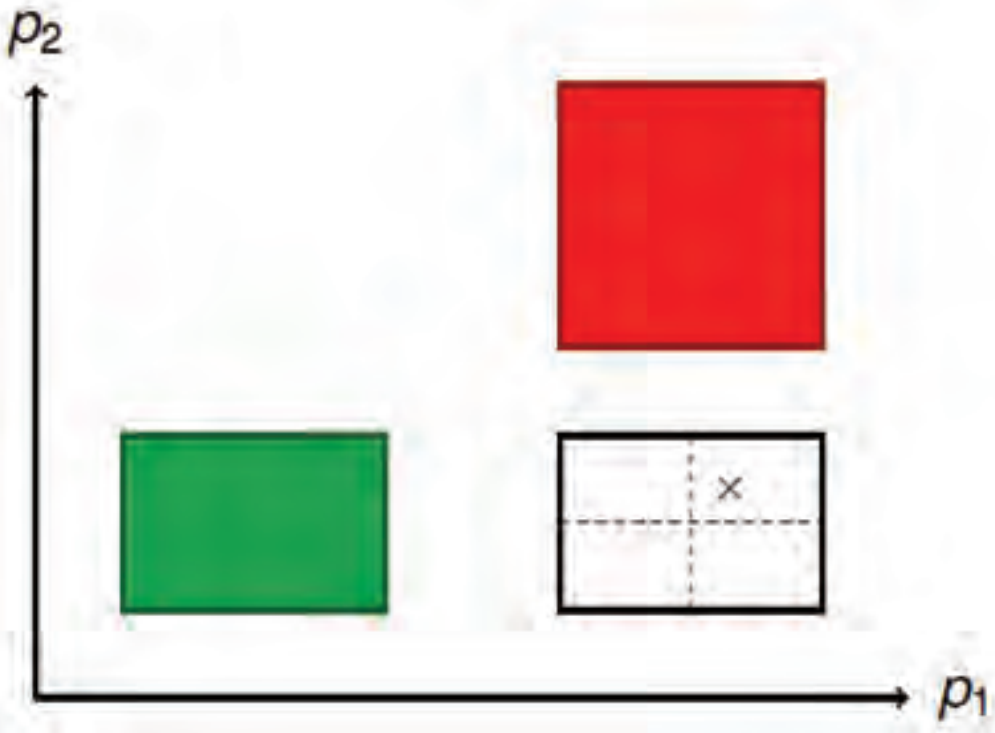
Idea: generate region candidates R and ask SMT solver[2] for counterexample

[2] Over non-linear real arithmetic using Z3 or SMT-RAT.

Exploiting SMT

Goal: partition parameter space in regions R that are either safe or unsafe

Idea: generate region candidates R and ask SMT solver[2] for counterexample



UNSAT $\Rightarrow$ R safe / unsafe

SAT $\Rightarrow$ resample & refine

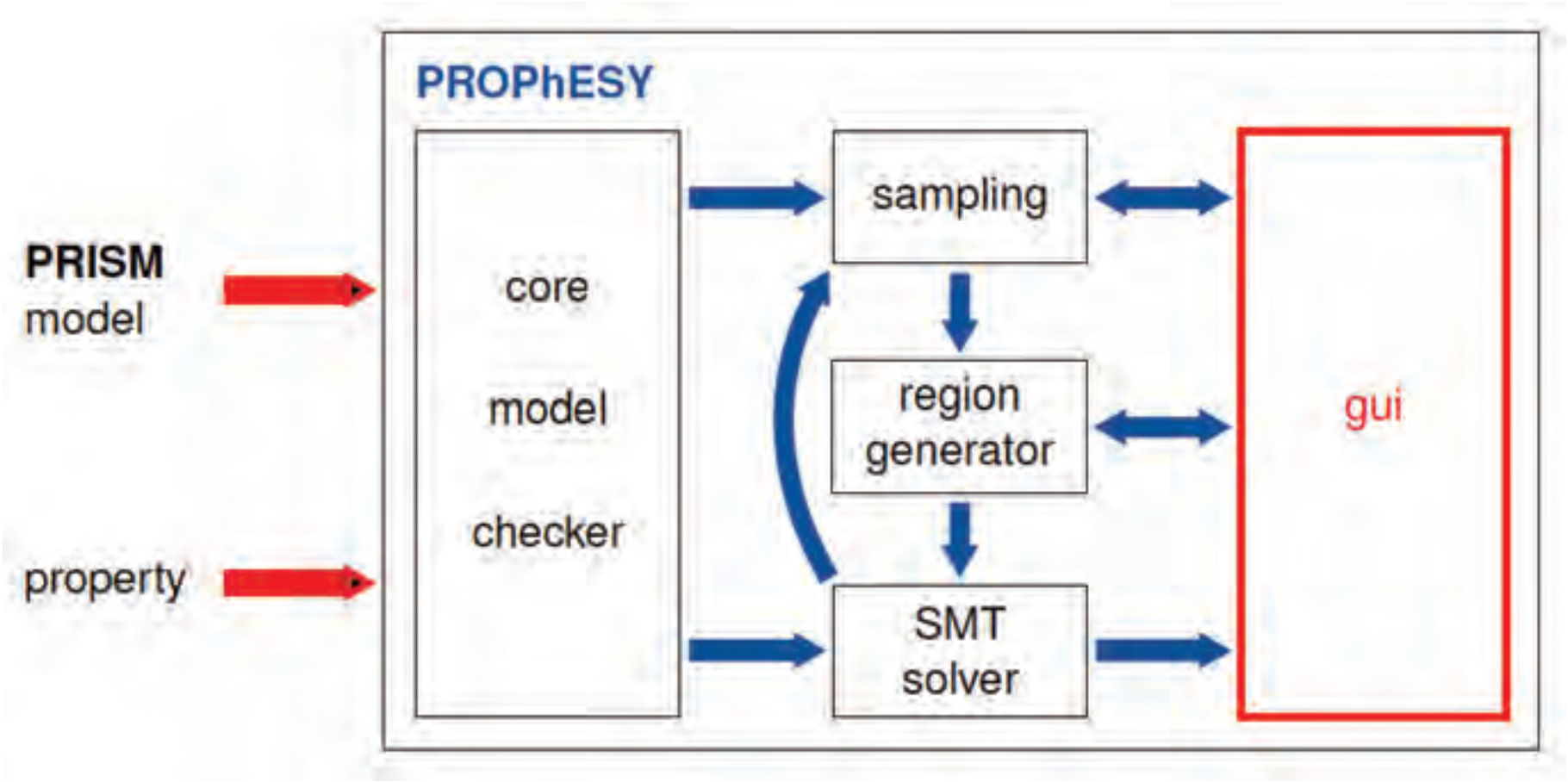
timeout $\Rightarrow$ smaller regions

strategies

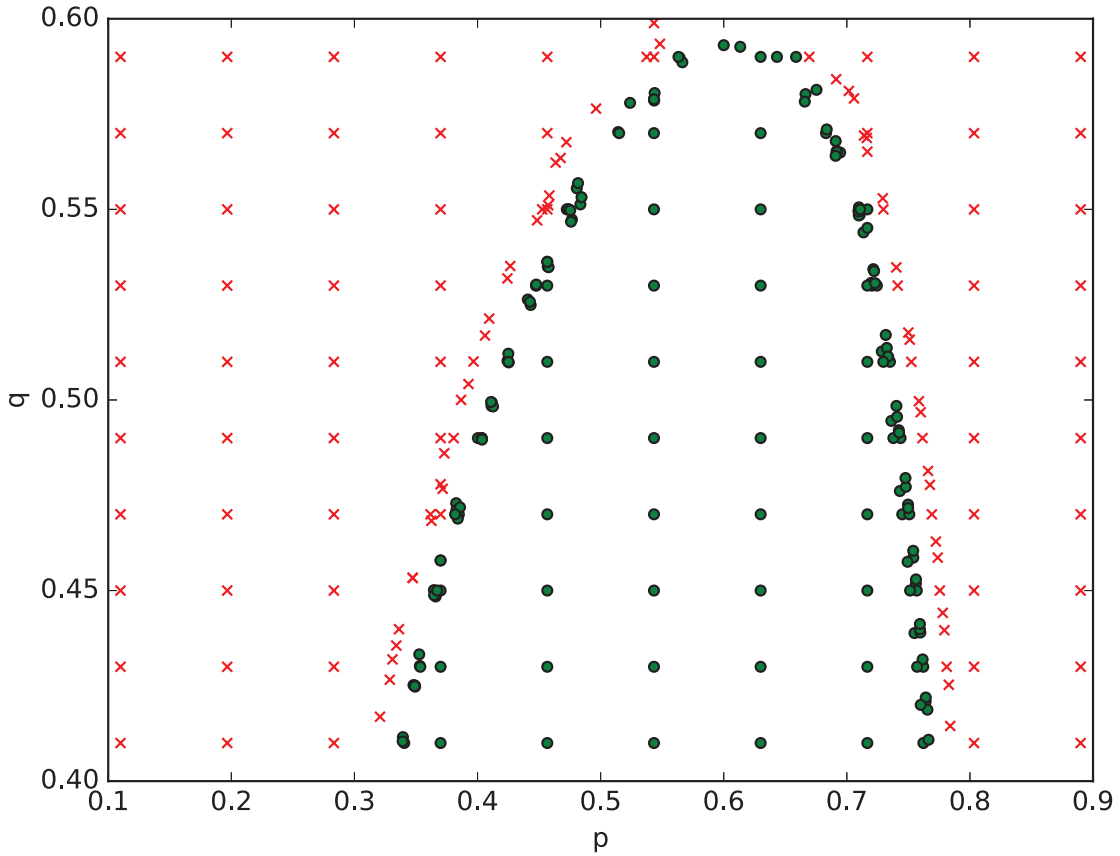
- cubes
- growing rectangles
- halfspaces

[2] Over non-linear real arithmetic using Z3 or SMT-RAT.

CEGAR-Like Parameter Synthesis

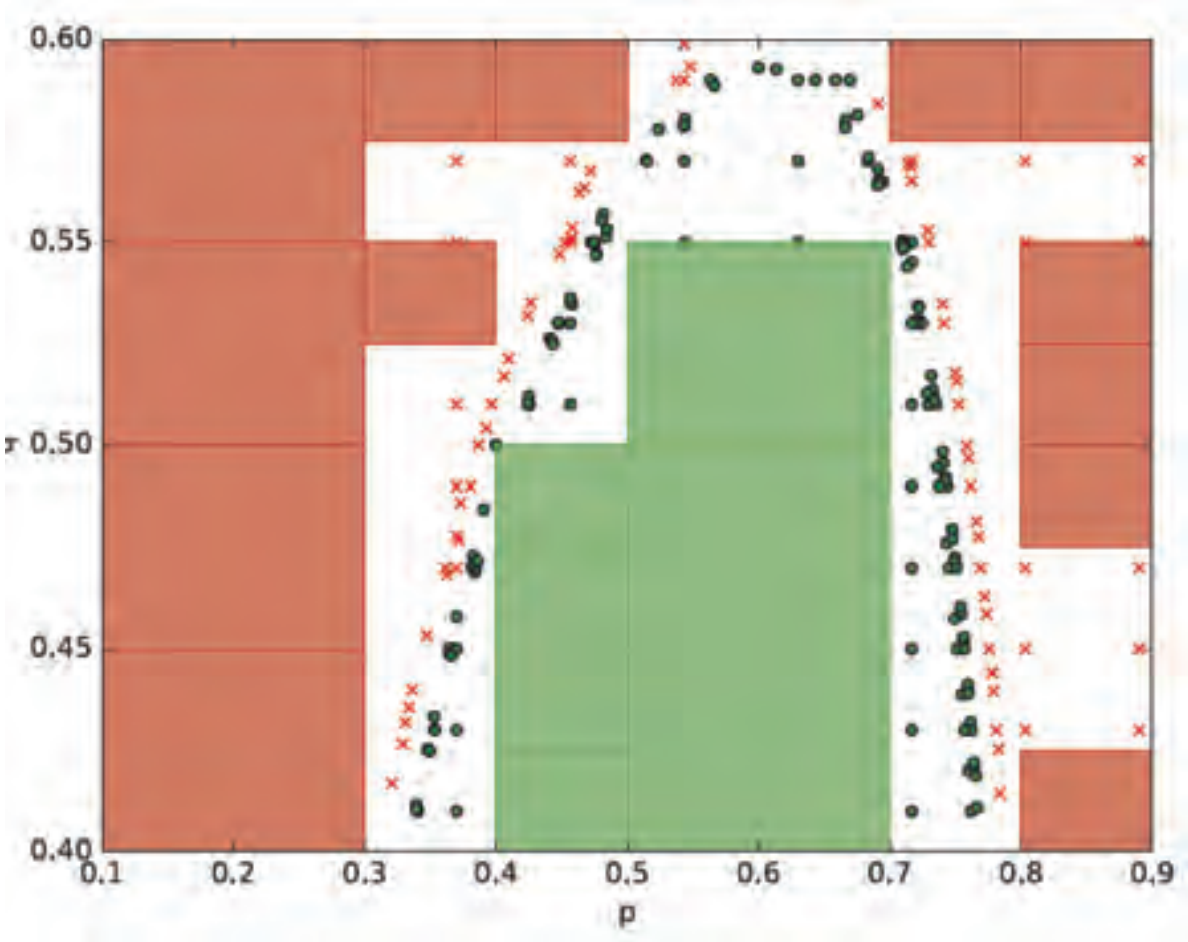


CEGAR-Like Parameter Synthesis



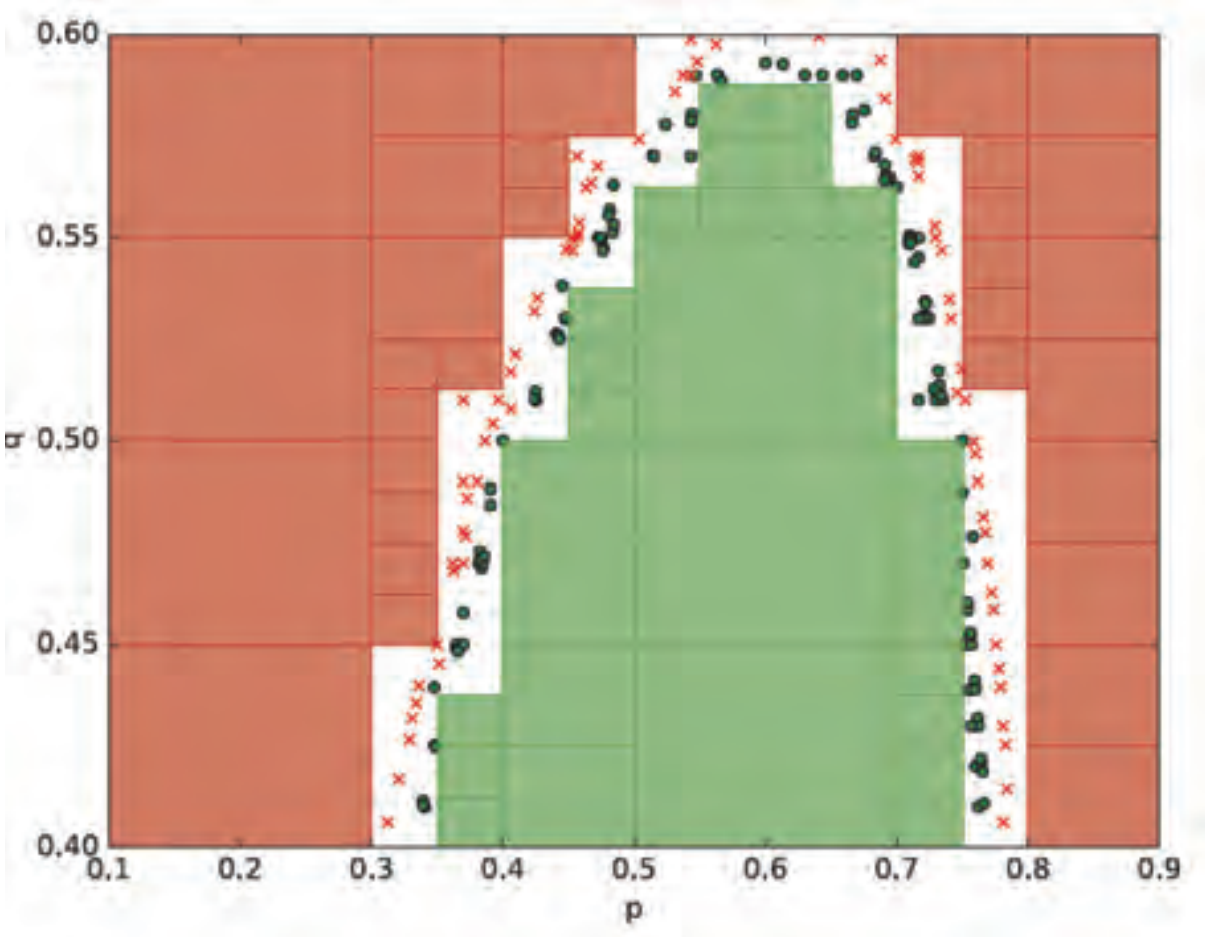
For which $1/10 \leqslant p \leqslant 9/10$ and $2/5 \leqslant q \leqslant 3/5$ does $\Pr(\Diamond 2) \geqslant 3/20$ hold?

CEGAR-Like Parameter Synthesis



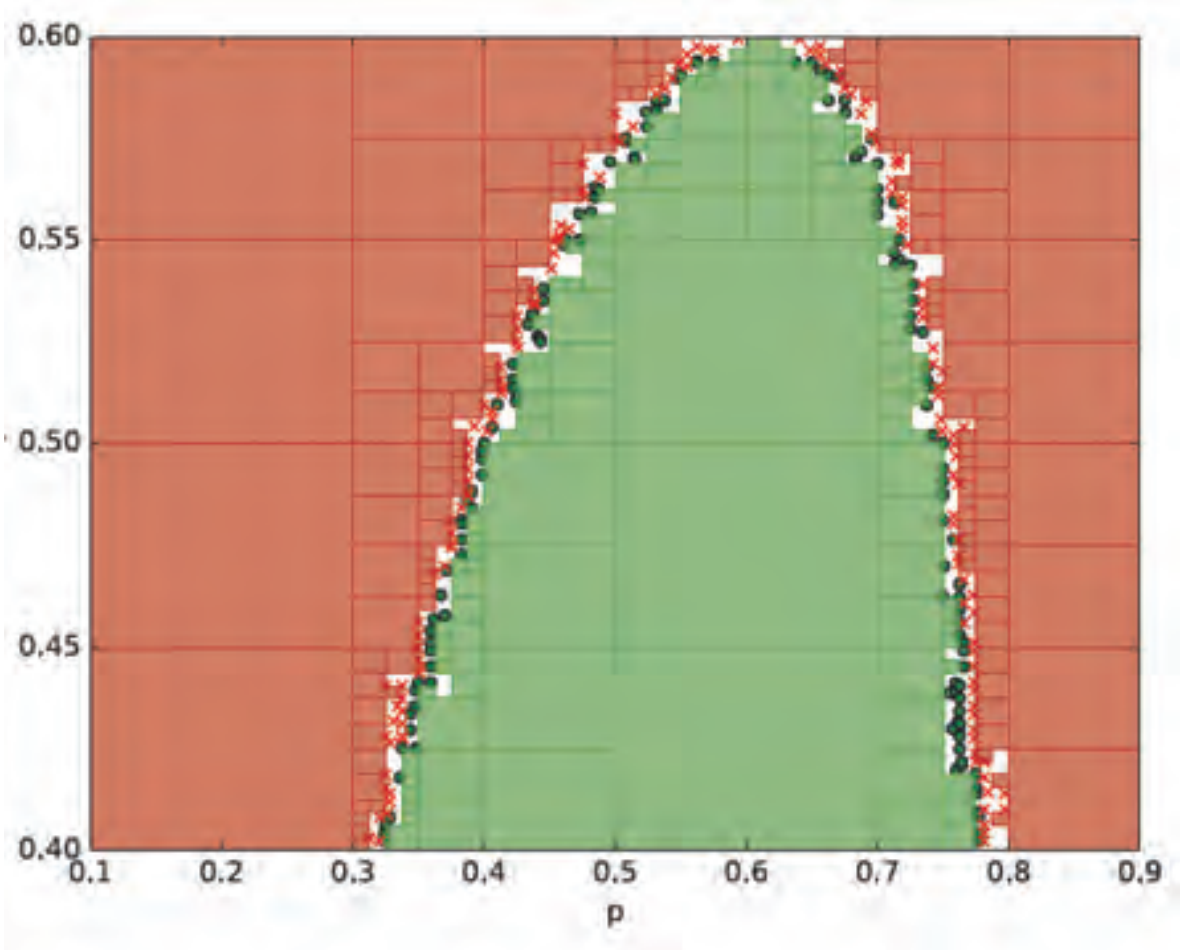
For which $1/10 \leqslant p \leqslant 9/10$ and $2/5 \leqslant q \leqslant 3/5$ does $\Pr(\Diamond 2) \geqslant 3/20$ hold?

CEGAR-Like Parameter Synthesis



For which $1/10 \leqslant p \leqslant 9/10$ and $2/5 \leqslant q \leqslant 3/5$ does $\Pr(\Diamond 2) \geqslant 3/20$ hold?

CEGAR-Like Parameter Synthesis



For which $1/10 \leqslant p \leqslant 9/10$ and $2/5 \leqslant q \leqslant 3/5$ does $\Pr(\Diamond 2) \geqslant 3/20$ hold?

Experimental Results

[Dehnert *et al.*, 2015]

competitors

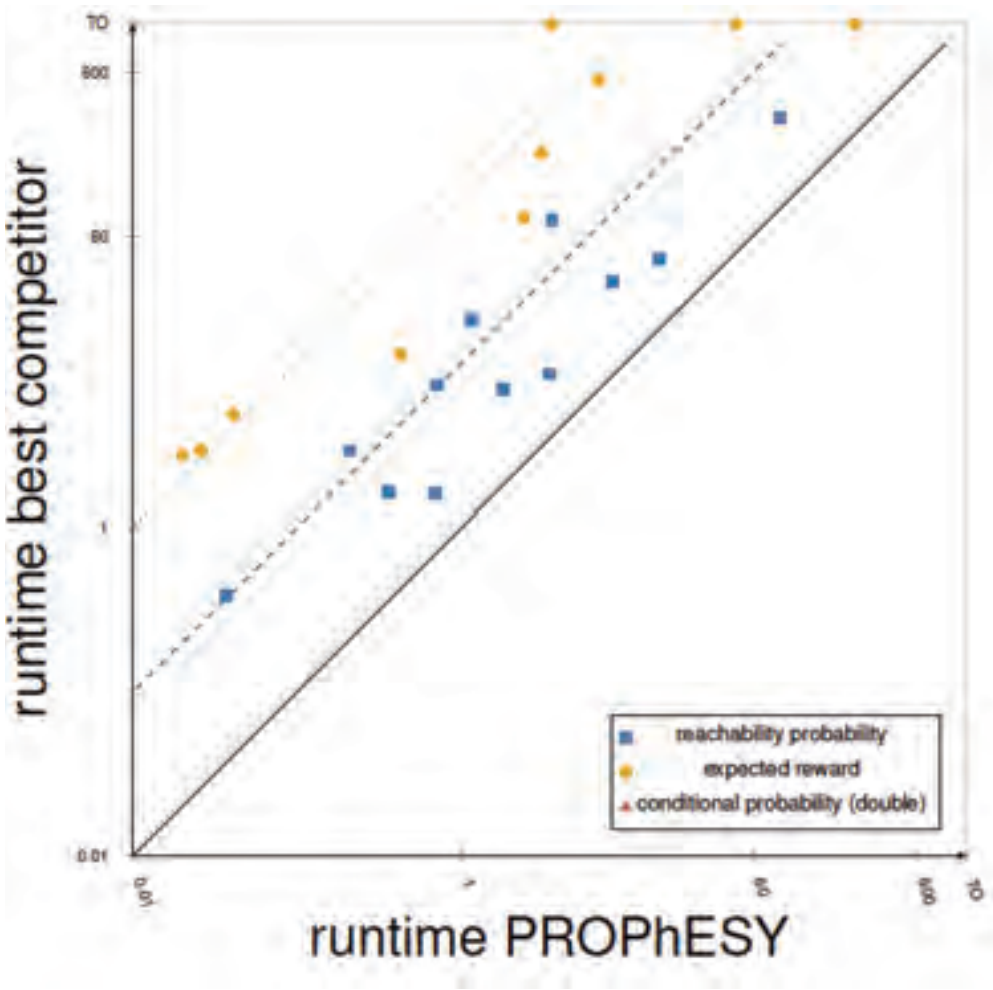
- `PARAM` [Hahn et al., 2010]
- `PRISM` [Parker et al., 2011]

models

- Bounded retransmission protocol
- NAND multiplexing
- Zeroconf, Crowds protocol
- 10^4 to $7.5 \cdot 10^6$ states

experiments:

- best set-up for each tool
- log-scale x- and y-axis



runner-up in the CAV 2015 artefact evaluation

Experimental Results

[Dehnert *et al.*, 2015]

competitors

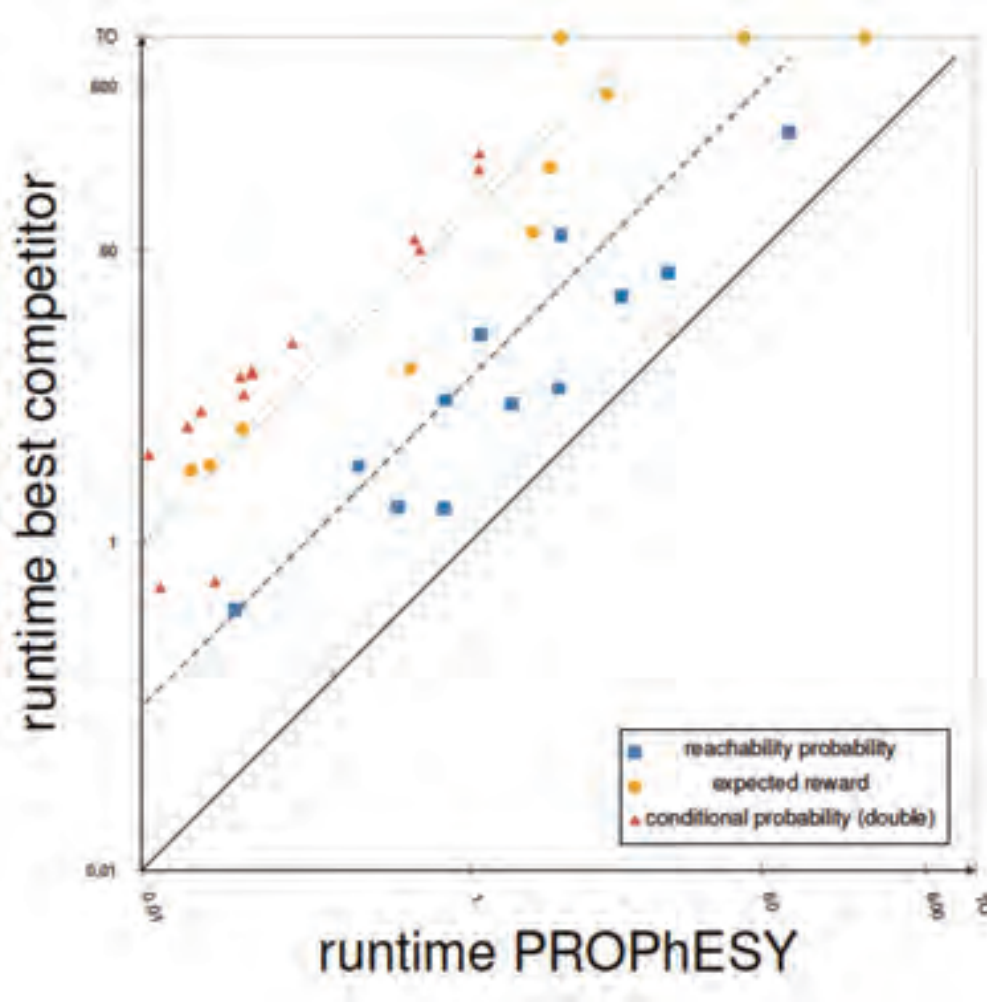
- PARAM [Hahn et al., 2010]
- PRISM [Parker et al., 2011]
- prototype [Baier et al., 2014]

models

- Bounded retransmission protocol
- NAND multiplexing
- Zeroconf, Crowds protocol
- 10^4 to $7.5 \cdot 10^6$ states

experiments:

- best set-up for each tool
- log-scale x- and y-axis



runner-up in the CAV 2015 artefact evaluation

Parameter Synthesis using SMT

Pros:

- Exact results: rational function is an exact symbolic object
- Drastic improvements over existing tools `PARAM` and `PRISM`
- User-friendly representation

Parameter Synthesis using SMT

Pros:

- Exact results: rational function is an exact symbolic object
- Drastic improvements over existing tools — `PARAM` and `PRISM`
- User-friendly representation

Cons:

- Rational function requires many gcd-computations — > 4 parameters?
- SMT performance unpredictable — heuristics hard

Parameter Synthesis using SMT

Pros:

- Exact results: rational function is an exact symbolic object
- Drastic improvements over existing tools — PARAM and PRISM
- User-friendly representation

Cons:

- Rational function requires many gcd-computations — > 4 parameters?
- SMT performance unpredictable — heuristics hard

Can we do better by sacrificing exactness? Yes.

Approximate Parameter Synthesis [Quatmann *et al*,, 2016]

Let transition probabilities be linear in each variable.

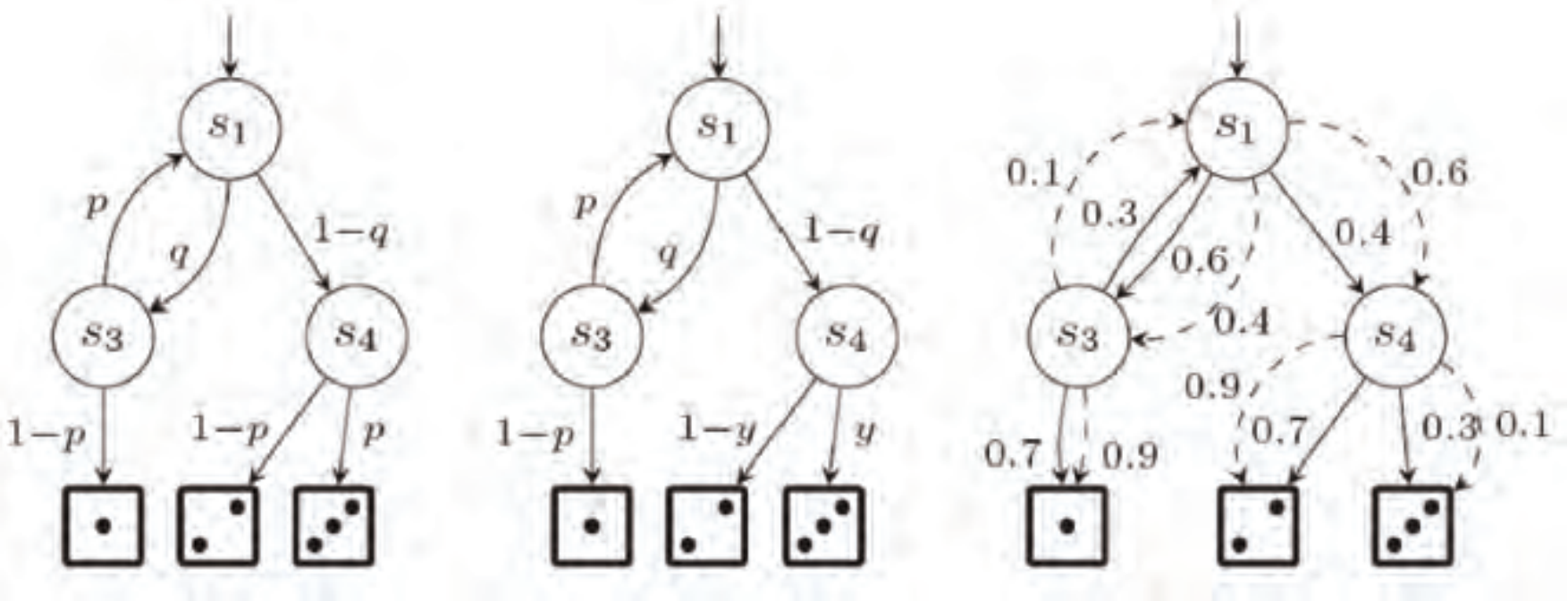
That is, transition functions f are multi-affine multivariate polynomials of form:

$$f = \sum a_i \cdot \left(\prod_{x \in V} x \right) \text{ with } a_i \in \mathbb{Q}$$

Examples: $3x{\cdot}y + 4y{\cdot}z$, $1 - x$, $x{\cdot}y{\cdot}z$ etc.

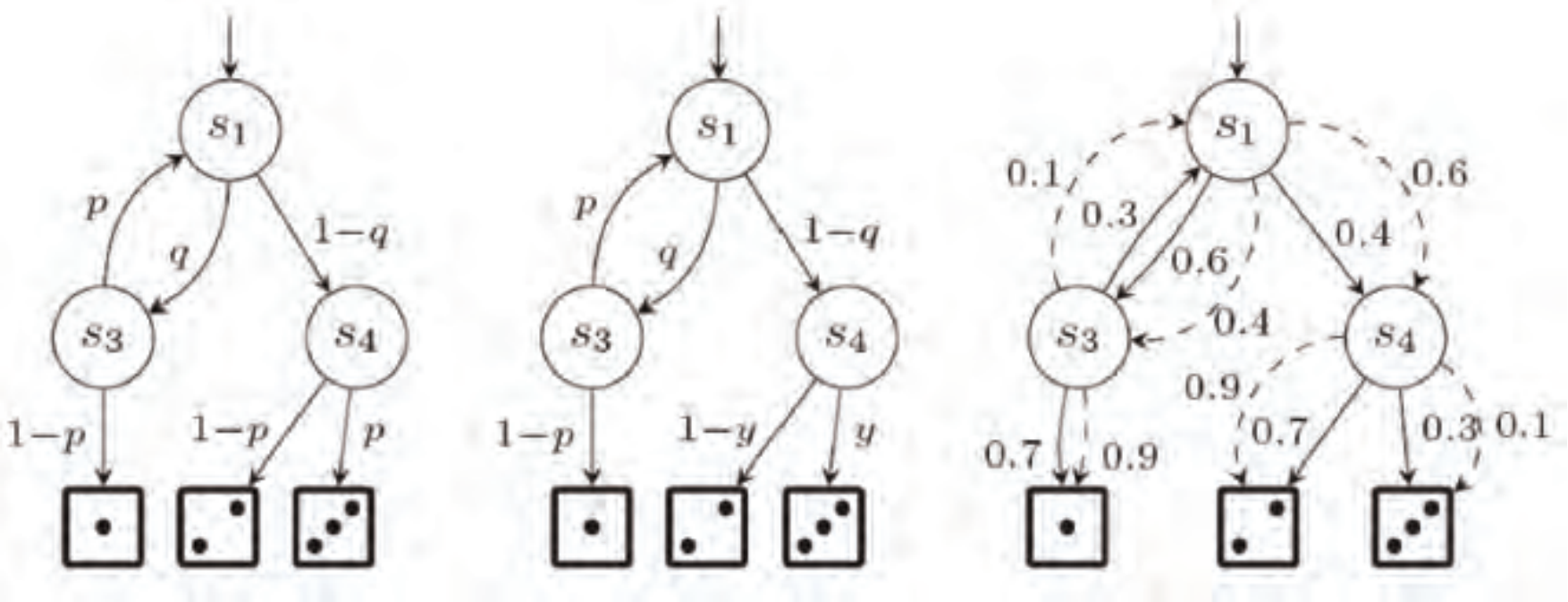
Approximate Parameter Synthesis [Quatmann *et al*,, 2016]

Approximate Parameter Synthesis [Quatmann *et al.*, 2016]



Two-phase approach: first remove dependencies, then substitute extremal values

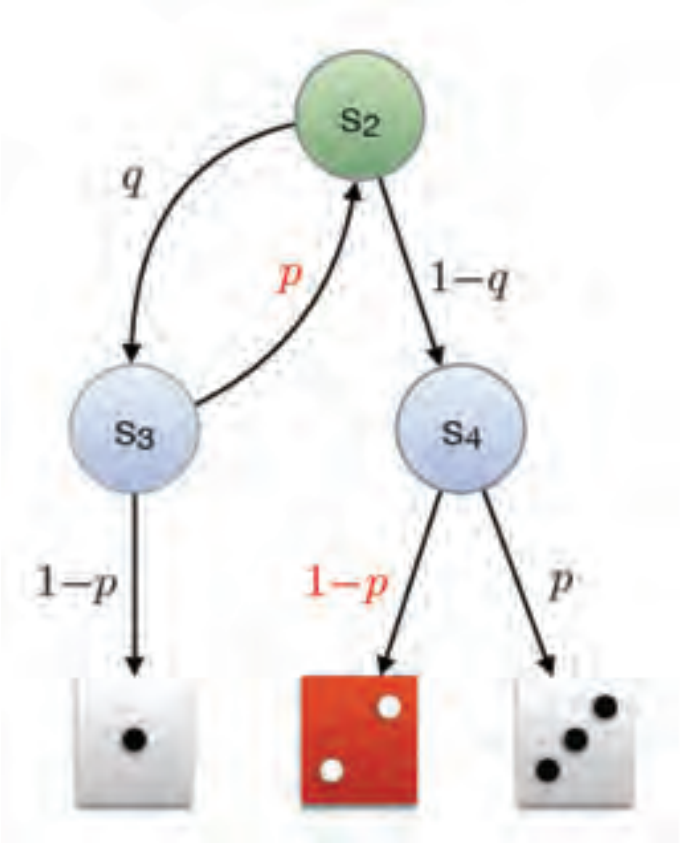
Approximate Parameter Synthesis [Quatmann *et al.*, 2016]

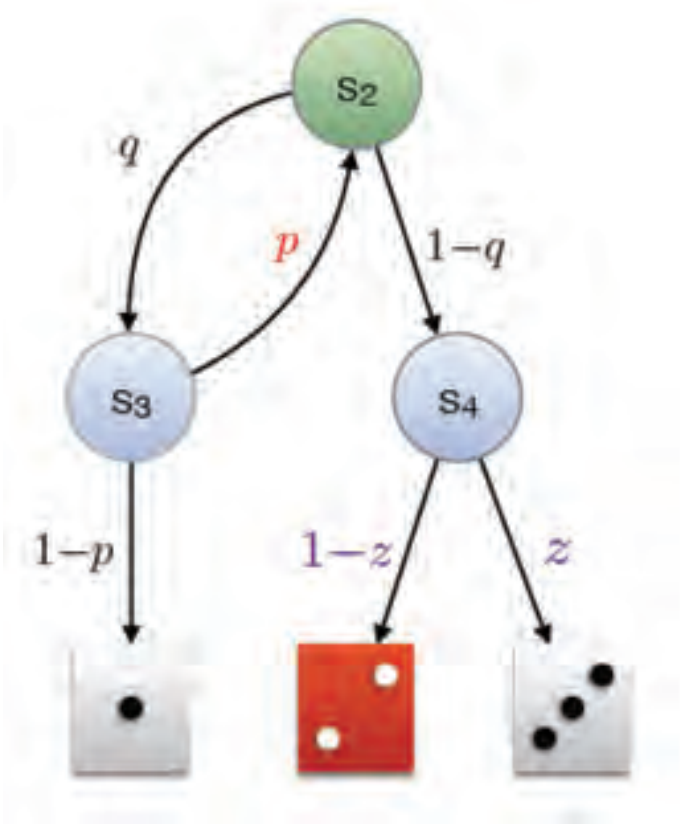


Two-phase approach: first remove dependencies, then substitute extremal values

Also applicable to parametric MDPs.

Phase 1: Relaxation





Parameter dependencies are removed; $\Pr(\Diamond 2) = (1 - z) \cdot \frac{1-q}{1-p \cdot q}$

$\Rightarrow$ each state is equipped with its own parameter

Phase 1: Relaxation

Correctness:

- Relaxed regions contain more valuations than original regions

$\Rightarrow$ Relaxation yields over-approximations

$\Rightarrow$ Relaxation preserves upper-bounds on reachability probs

Phase 1: Relaxation

Correctness:

- Relaxed regions contain more valuations than original regions

$\Rightarrow$ Relaxation yields over-approximations

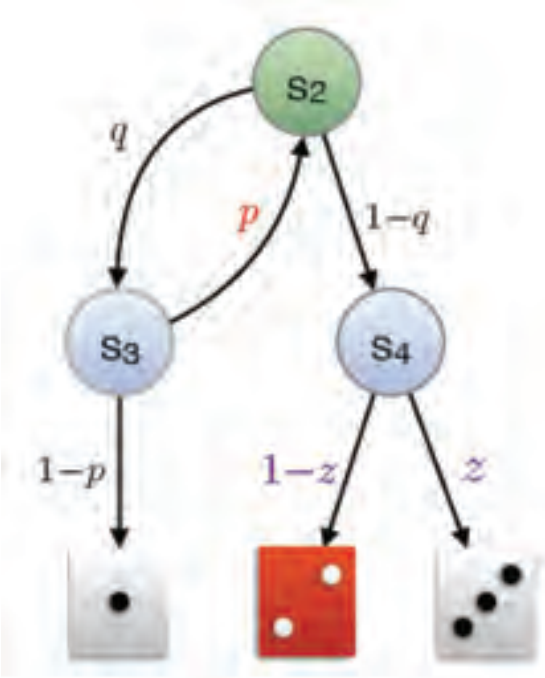
$\Rightarrow$ Relaxation preserves upper-bounds on reachability probs

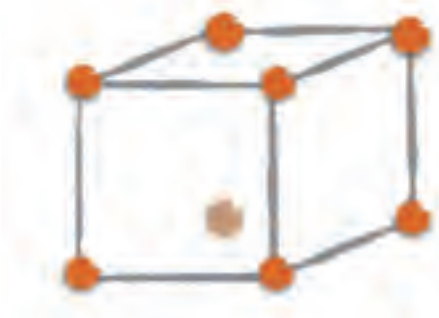
Complexity of parameter synthesis :

- Relaxation increases the number of parameters
- Extremal values of the state parameters attain maximal probabilities

$\Rightarrow$ Valuations for maximal probabilities are easier to find

Phase 2: Substitution



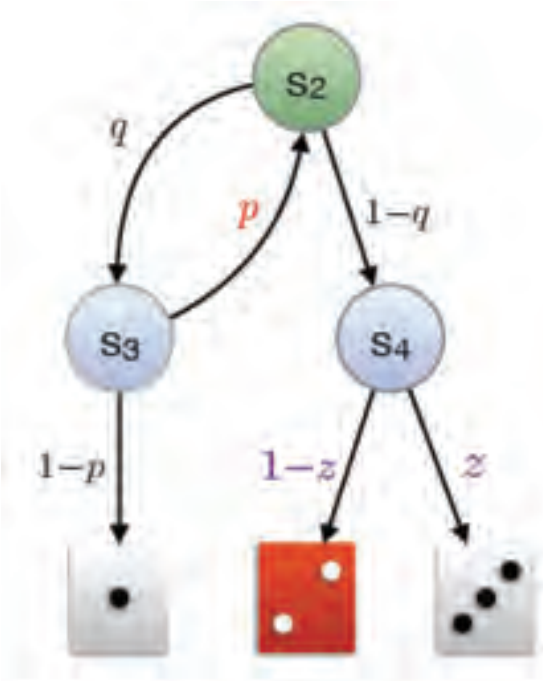


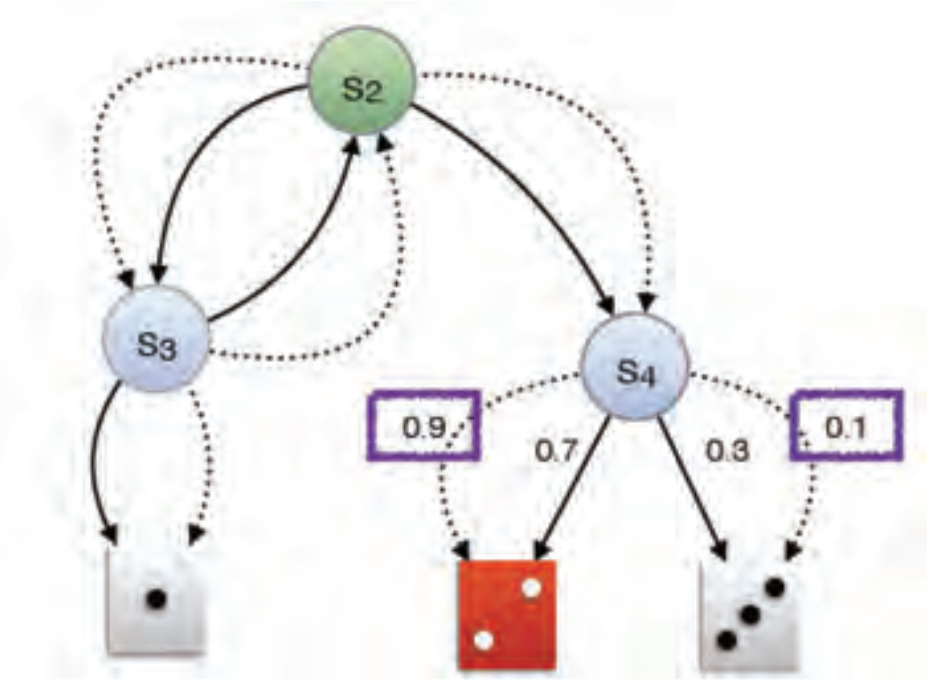
$$p, z \in [0.1, 0.3] \qquad q \in [0.4, 0.6]$$

$$p, z \in \{0.1, 0.3\} \qquad q \in \{0.4, 0.6\}$$

Local parameters per state $\Rightarrow$ extremal values at states suffice

Phase 2: Substitution



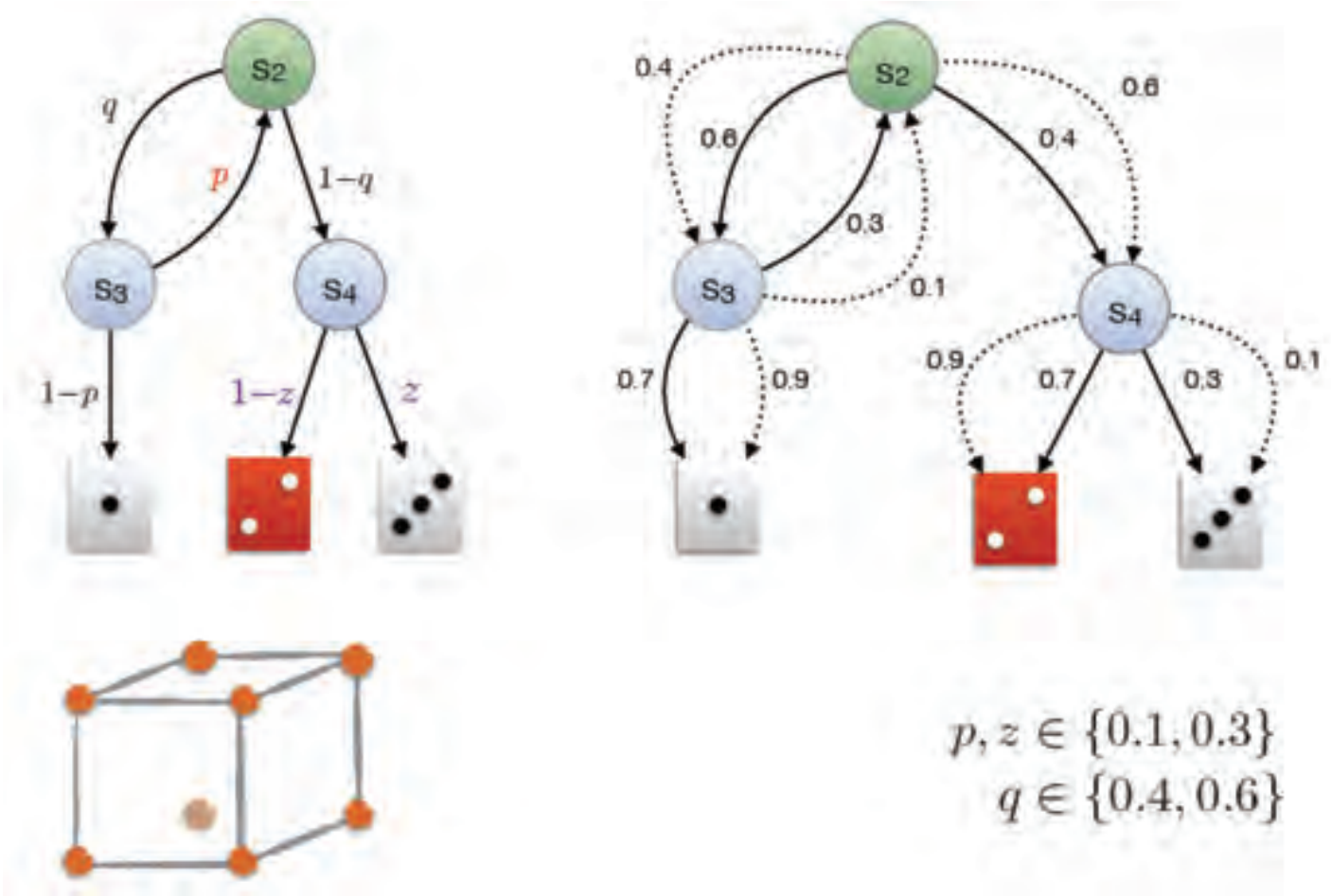




$$p, z \in \{0.1, 0.3\}$$
$$q \in \{0.4, 0.6\}$$

Local parameters per state ⇒ extremal values at states suffice

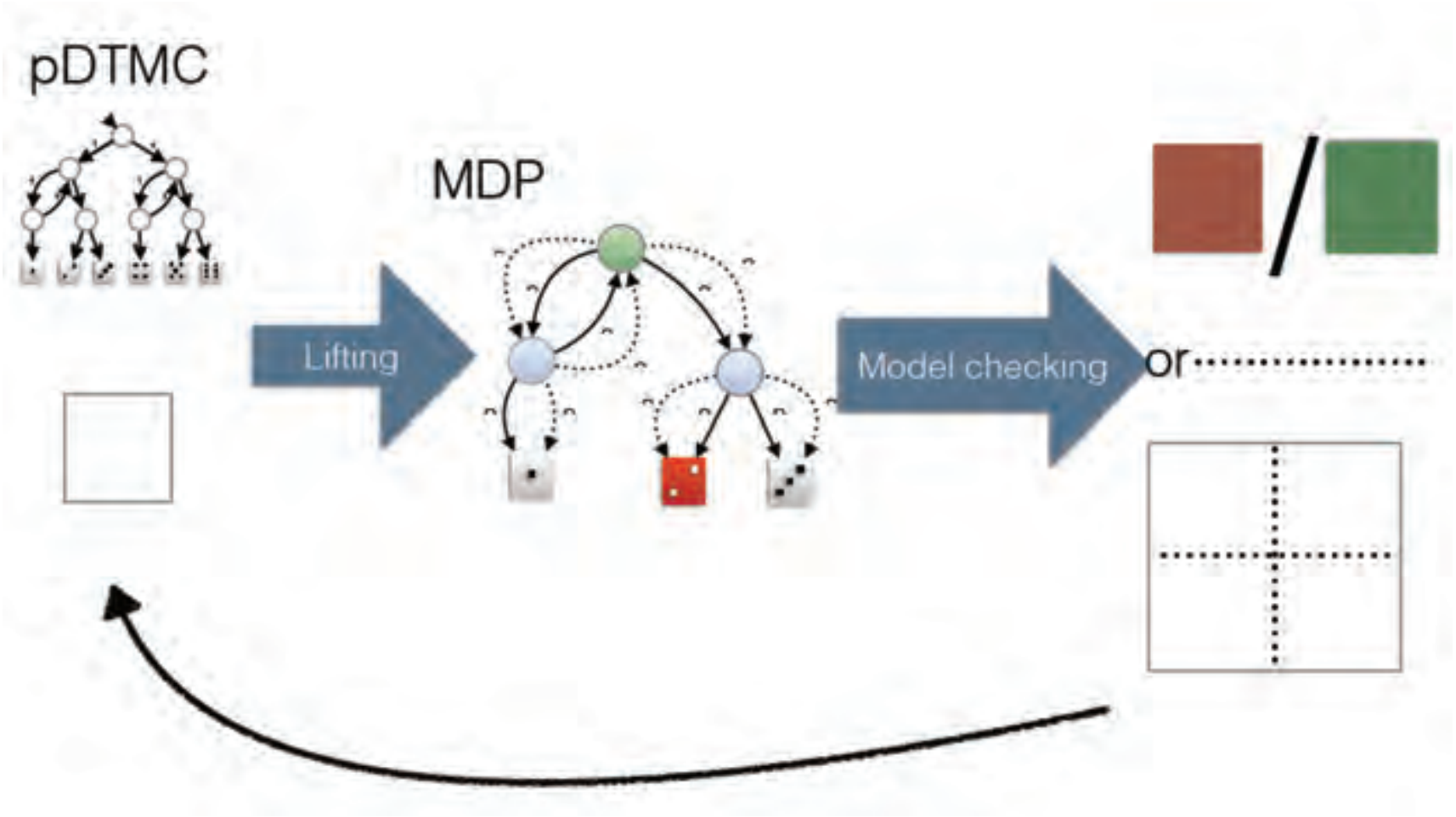
Phase 2: Substitution



This results in a Markov decision process.

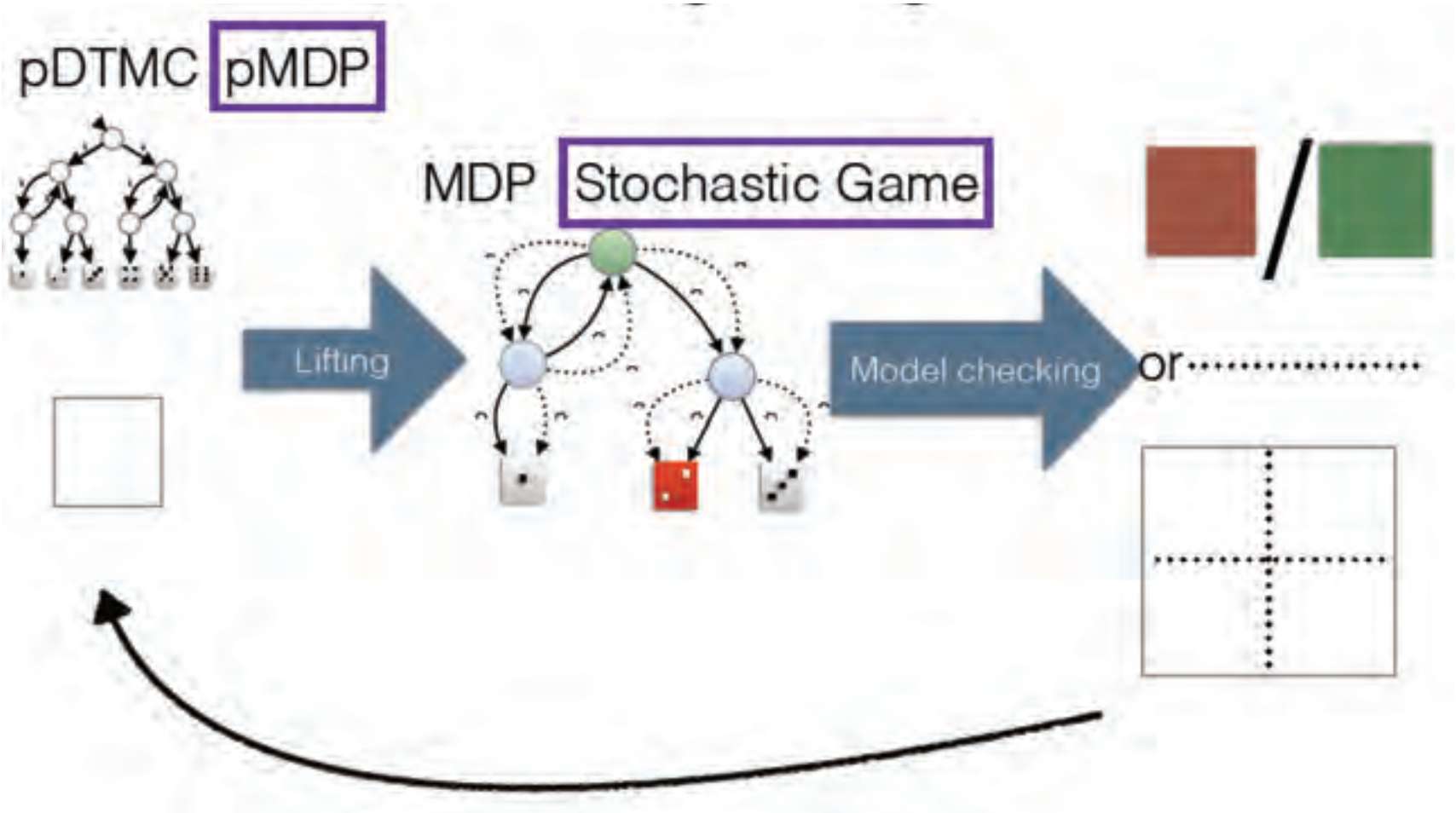
Its extremal reachability probabilities provide bounds for parametric MC.

Parameter Synthesis



Until ≈ 95% of the parameter space is covered

Parameter Synthesis



Until 95% of the parameter space is covered

Coverage

		φ	n	# states	# trans	% p	t	safe	unsafe	neither	unkn
pMC	brp	$\mathbb{E}$	2	20 744	27 651	48%	**51**	14.9%	79.2%	**5.8%**	**0.2%**
		$\mathbb{E}$	4	20 744	27 651	48%	**71**	7.5%	51.0%	**40.6%**	**0.8%**
	crowds	$\mathbb{P}$	2	104 512	246 082	19%	**44**	54.4%	41.1%	**4.2%**	**0.3%**
	nand	$\mathbb{P}$	2	35 112	52 647	47%	**21**	21.4%	68.5%	**6.9%**	**3.2%**
pMDP	brp	$\mathbb{P}$	2	40 721	55 143	50%	**153**	6.6%	90.4%	**3.0%**	**0.0%**
	cons	$\mathbb{P}$	4	22 656	75 232	41%	**357**	2.6%	87.0%	**10.4%**	**0.0%**
	sav	$\mathbb{P}$	4	379	1 127	50%	**2**	44.0%	15.4%	**35.4%**	**5.3%**
	zconf	$\mathbb{P}$	2	88 858	203 550	40%	**186**	16.6%	77.3%	**5.6%**	**0.5%**

Parameter space $R = [10^{-5}, 1-10^{-5}]^n$ until 95% coverage for n parameters for 625 equally-sized regions without region refinement

single core, 2.0 GHz, 30GB RAM, TO = one hour

Parametric Markov Chain Benchmarks

						PLA			PRISM
benchmark	instance	φ	#pars	#states	#trans	#regions	direct	bisim	best
brp	(256,5)	$\mathbb{P}$	2	19 720	26 627	37	**6**	14	TO
	(4096,5)	$\mathbb{P}$	2	315 400	425 987	13	**233**	TO	TO
	(256,5)	$\mathbb{E}$	2	20 744	27 651	195	**8**	15	TO
	(4096,5)	$\mathbb{E}$	2	331 784	442 371	195	502	**417**	TO
	(16,5)	$\mathbb{E}$	4	1 304	1 731	1 251 220	2 764	**1 597**	TO
	(32,5)	$\mathbb{E}$	4	2 600	3 459	1 031 893	TO	**2 722**	TO
	(256,5)	$\mathbb{E}$	4	20 744	27 651	–	TO	TO	TO
crowds	(10,5)	$\mathbb{P}$	2	104 512	246 082	123	17	**6**	2038
	(15,7)	$\mathbb{P}$	2	8 364 409	25 108 729	116	1 880	**518**	TO
	(20,7)	$\mathbb{P}$	2	45 421 597	164 432 797	119	TO	**2 935**	TO
nand	(10,5)	$\mathbb{P}$	2	35 112	52 647	469	**22**	30	TO
	(25,5)	$\mathbb{P}$	2	865 592	1 347 047	360	**735**	2 061	TO

coverage of 95%; refinement into four equally-sized regions

SMT approach needs >one hour on all instances.

Parametric MDP Benchmarks

						PLA			PRISM
benchmark	instance	φ	#pars	#states	#trans	#regions	direct	bisim	best
brp	(256,5)	$\mathbb{P}$	2	40 721	55 143	37	**35**	3 359	TO
	(4096,5)	$\mathbb{P}$	2	647 441	876 903	13	**3 424**	TO	TO
consensus	(2,2)	$\mathbb{P}$	2	272	492	119	< **1**	< **1**	31
	(2,32)	$\mathbb{P}$	2	4 112	7 692	108	**113**	141	TO
	(4,2)	$\mathbb{P}$	4	22 656	75 232	6 125	**1 866**	2 022	TO
	(4,4)	$\mathbb{P}$	4	43 136	144 352	–	TO	TO	TO
sav	(6,2,2)	$\mathbb{P}$	2	379	1 127	162	< **1**	< **1**	TO
	(100,10,10)	$\mathbb{P}$	2	1 307 395	6 474 535	37	**1 612**	TO	TO
	(6,2,2)	$\mathbb{P}$	4	379	1 127	621 175	944	**917**	TO
	(10,3,3)	$\mathbb{P}$	4	1 850	6 561		TO	TO	TO
zeroconf	(2)	$\mathbb{P}$	2	88 858	203 550	186	**86**	1 295	TO
	(5)	$\mathbb{P}$	2	494 930	1 133 781	403	**2 400**	TO	TO

coverage of 95%

Summary So Far

SMT-based approach:

- Exact
- Requires rational functions
- Fickle SMT performance
- $\approx 10^6$ states, 2 parameters
- Restricted to Markov chains
- CEGAR-like refinement

Summary So Far

SMT-based approach:

- Exact
- Requires rational functions
- Fickle SMT performance
- $\approx 10^6$ states, 2 parameters
- Restricted to Markov chains
- CEGAR-like refinement

Parameter lifting approach:

- Approximative
- Off-the-shelf model checking
- No SMT, no rational functions
- $\approx 10^7$ states, 4–5 parameters
- Applicable to MDPs and games
- CEGAR-like refinement

Multiple Objectives

Multiple Objectives

Inputs:

1. a (finite) parametric MDP $\mathcal{M}$ over $V = \{ x_1, \ldots, x_n \}$
   with signomial parameter functions $c \cdot x_1^{a_1} \cdot \ldots \cdot x_n^{a_n}$ for $c \in \mathbb{R}$
2. multiple objectives $\varphi_1, \ldots, \varphi_m$ (reachability, expected reward)
3. objective function f over V: $\sum_{k=1}^{N} c_k \cdot x_1^{a_{1k}} \cdot \ldots \cdot x_n^{a_{nk}}$ for $c_k \in \mathbb{R}$

Multiple Objectives

Inputs:

1. a (finite) parametric MDP $\mathcal{M}$ over $V = \{ x_1, \ldots, x_n \}$
   with signomial parameter functions $c \cdot x_1^{a_1} \cdot \ldots \cdot x_n^{a_n}$ for $c \in \mathbb{R}$
2. multiple objectives $\varphi_1, \ldots, \varphi_m$ (reachability, expected reward)
3. objective function f over V: $\sum_{k=1}^{N} c_k \cdot x_1^{a_{1k}} \cdot \ldots \cdot x_n^{a_{nk}}$ for $c_k \in \mathbb{R}$

Output:

A (randomised) policy σ and valuation u such that:

$$\underbrace{\mathcal{M}^\sigma[u] \vDash \varphi_1 \wedge \ldots \wedge \varphi_m}_{\text{“feasibility”}} \quad \text{and} \quad \underbrace{\text{the objective } f \text{ is minimised}}_{\text{“optimality”}}$$

Multiple Objectives

Inputs:

1. a (finite) parametric MDP $\mathcal{M}$ over $V = \{ x_1, \ldots, x_n \}$
   with signomial parameter functions $c \cdot x_1^{a_1} \cdot \ldots \cdot x_n^{a_n}$ for $c \in \mathbb{R}$
2. multiple objectives $\varphi_1, \ldots, \varphi_m$ (reachability, expected reward)
3. objective function f over V: $\sum_{k=1}^{N} c_k \cdot x_1^{a_{1k}} \cdot \ldots \cdot x_n^{a_{nk}}$ for $c_k \in \mathbb{R}$

Output:

A (randomised) policy σ and valuation u such that:

$$\underbrace{\mathcal{M}^\sigma[u] \vDash \varphi_1 \wedge \ldots \wedge \varphi_m}_{\text{“feasibility”}} \quad \text{and} \quad \underbrace{\text{the objective } f \text{ is minimised}}_{\text{“optimality”}}$$

multi-objective MDP: use LP [Etessami *et al.*, 2008]

multi-objective parametric MDP: use special type NLP [Cubuktepe *et al.*, 2017]

NLP for Two Objectives

Objectives: minimise f, reach T with probability $\leqslant p$, expected cost to reach $G \leqslant c$

Subject to: $p_{s_I} \leqslant p$ reachability objective

$c_{s_I} \leqslant c$ expected reward objective

NLP for Two Objectives

Objectives: minimise f, reach T with probability $\leqslant p$, expected cost to reach $G \leqslant c$

Subject to: $p_{s_I} \leqslant p$ reachability objective

$c_{s_I} \leqslant c$ expected reward objective

$\forall s: \quad \sum_{\alpha \in Act(s)} \sigma^{s,\alpha} = 1$ randomised scheduler

$\forall s, \alpha: \quad 0 \leqslant \sigma^{s,\alpha} \leqslant 1$

NLP for Two Objectives

Objectives: minimise f, reach T with probability $\leqslant p$, expected cost to reach $G \leqslant c$

Subject to:

$$p_{s_I} \leqslant p \qquad \text{reachability objective}$$

$$c_{s_I} \leqslant c \qquad \text{expected reward objective}$$

$$\forall s: \quad \sum_{\alpha \in Act(s)} \sigma^{s,\alpha} = 1 \qquad \text{randomised scheduler}$$

$$\forall s, \alpha: \quad 0 \leqslant \sigma^{s,\alpha} \leqslant 1$$

$$\forall s, \alpha: \quad \sum_{t \in S} \mathcal{P}(s, \alpha, t) = 1 \qquad \text{probabilistic choice}$$

$$\forall s, t, \alpha: \quad 0 \leqslant \mathcal{P}(s, \alpha, t) \leqslant 1$$

NLP for Two Objectives

Objectives: minimise f, reach T with probability $\leqslant p$, expected cost to reach $G \leqslant c$

Subject to:

$$p_{s_I} \leqslant p \qquad \text{reachability objective}$$

$$c_{s_I} \leqslant c \qquad \text{expected reward objective}$$

$$\forall s: \quad \sum_{\alpha \in Act(s)} \sigma^{s,\alpha} = 1 \qquad \text{randomised scheduler}$$

$$\forall s, \alpha: \quad 0 \leqslant \sigma^{s,\alpha} \leqslant 1$$

$$\forall s, \alpha: \quad \sum_{t \in S} \mathcal{P}(s, \alpha, t) = 1 \qquad \text{probabilistic choice}$$

$$\forall s, t, \alpha: \quad 0 \leqslant \mathcal{P}(s, \alpha, t) \leqslant 1$$

$$\forall s \in T: \quad p_s = 1 \qquad \text{reach prob of } T$$

$$\forall s \notin T: \quad p_s = \sum_{\alpha \in Act(s)} \sigma^{s,\alpha} \cdot \sum_{t \in S} \mathcal{P}(s, \alpha, t) \cdot p_t \qquad \text{transition probabilities}$$

NLP for Two Objectives

Objectives: minimise f, reach T with probability $\leqslant p$, expected cost to reach $G \leqslant c$

Subject to:

$$p_{s_I} \leqslant p \qquad \text{reachability objective}$$

$$c_{s_I} \leqslant c \qquad \text{expected reward objective}$$

$$\forall s: \quad \sum_{\alpha \in Act(s)} \sigma^{s,\alpha} = 1 \qquad \text{randomised scheduler}$$

$$\forall s, \alpha: \quad 0 \leqslant \sigma^{s,\alpha} \leqslant 1$$

$$\forall s, \alpha: \quad \sum_{t \in S} \mathcal{P}(s, \alpha, t) = 1 \qquad \text{probabilistic choice}$$

$$\forall s, t, \alpha: \quad 0 \leqslant \mathcal{P}(s, \alpha, t) \leqslant 1$$

$$\forall s \in T: \quad p_s = 1 \qquad \text{reach prob of } T$$

$$\forall s \notin T: \quad p_s = \sum_{\alpha \in Act(s)} \sigma^{s,\alpha} \cdot \sum_{t \in S} \mathcal{P}(s, \alpha, t) \cdot p_t \qquad \text{transition probabilities}$$

$$\forall s \in G: \quad c_s = 0 \qquad \text{expected cost of } G$$

$$\forall s \notin G: \quad c_s = \sum_{\alpha \in Act(s)} \sigma^{s,\alpha} \cdot \left(c(s, \alpha) + \sum_{t \in S} \mathcal{P}(s, \alpha, t) \cdot c_t \right) \qquad \text{expected costs}$$

Theorem: This NLP is sound and complete. But solving NLPs is exponential.

Can We Do Better?



Yes.

1. Get a feasible solution in polynomial time[3]. How? Geometric programming.
2. Get local optimum. How? Sequential convex programming.

Solutions are approximations that can be arbitrarily close.

[3] Approximation of arbitrarily precise results by interior point methods with barriers

Geometric Programming

Objective: minimise $f :: \sum_{k=1}^{N} c_k \cdot x_1^{a_{1k}} \cdot \ldots \cdot x_n^{a_{nk}}$ for $c_k \in \mathbb{R}_{\geqslant 0}$

Subject to:

$$\forall i \in [1..m]: \quad g_i \leqslant 1 \qquad \text{posynomial } g_i$$

$$\forall j \in [1..\ell]: \quad h_j = 1 \qquad \text{monomial } h_j$$

Division transformation: $f \leqslant h$ if and only if $\frac{f}{h} \leqslant 1$

Relaxation: $f = h$ implies $f \leqslant h$ if and only if $\frac{f}{h} \leqslant 1$

Convexification

$$\forall s \in S \setminus T. \quad p_s = \sum_{\alpha \in Act(s)} \sigma^{s,\alpha} \cdot \sum_{s' \in S} \mathcal{P}(s,\alpha,s') \cdot p_{s'}$$

upper bound on actual probability

⇓ division transformation

$$\forall s \in S \setminus T. \quad p_s \geq \sum_{\alpha \in Act(s)} \sigma^{s,\alpha} \cdot \sum_{s' \in S} \mathcal{P}(s,\alpha,s') \cdot p_{s'}$$

⇓ relaxation

still signomials (not convex)

$$\forall s \in S \setminus T. \quad \frac{\sum_{\alpha \in Act(s)} \sigma^{s,\alpha} \cdot \sum_{s' \in S} \mathcal{P}(s,\alpha,s') \cdot p_{s'}}{p_s} \leq 1$$

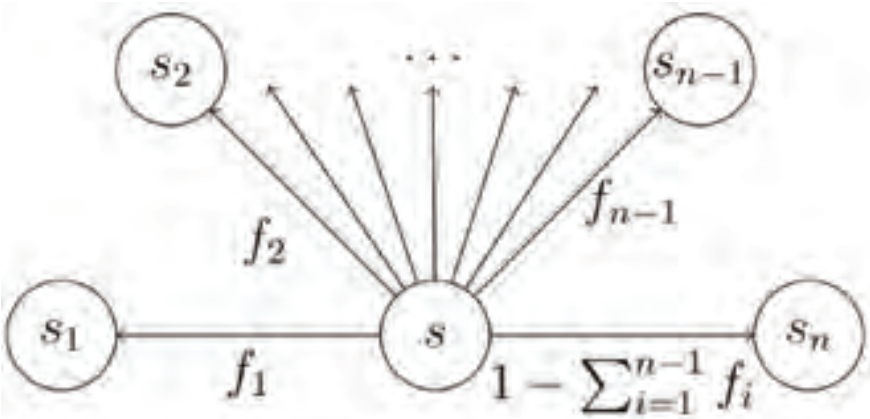
Lifting

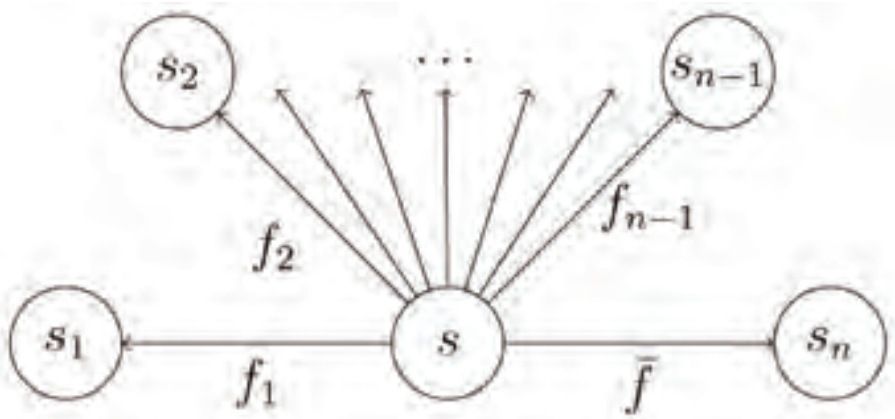
$$\mathcal{P}(s,\alpha,\bar{s}) = 1 - \sum_{s' \in S \setminus \{\bar{s}\}} \mathcal{P}(s,\alpha,s') \Longrightarrow \bar{\mathcal{P}}(s,\alpha,\bar{s}) = \bar{p}_{s,\alpha,\bar{s}} \in L$$

signomial

posynomial

lifting variable





GP for Two Objectives

Objectives: reach T with probability $\leqslant p$, expected cost to reach $G \leqslant c$

Subject to:

$$\frac{p_{s_I}}{p} \leqslant 1 \qquad \text{reachability}$$

$$\frac{c_{s_I}}{c} \leqslant 1 \qquad \text{expected reward}$$

GP for Two Objectives

Objectives: reach T with probability $\leqslant p$, expected cost to reach $G \leqslant c$

Subject to:

$$\frac{p_{s_I}}{p} \leqslant 1 \qquad \text{reachability}$$

$$\frac{c_{s_I}}{c} \leqslant 1 \qquad \text{expected reward}$$

$$\forall s: \quad \sum_{\alpha \in Act(s)} \sigma^{s,\alpha} \leqslant 1 \qquad \text{randomised scheduler}$$

$$\forall s, \alpha: \quad \sigma^{s,\alpha} \leqslant 1$$

GP for Two Objectives

Objectives: reach T with probability $\leqslant p$, expected cost to reach G $\leqslant c$

Subject to:

$$\frac{p_{s_I}}{p} \leqslant 1 \qquad \text{reachability}$$

$$\frac{c_{s_I}}{c} \leqslant 1 \qquad \text{expected reward}$$

$$\forall s: \quad \sum_{\alpha \in Act(s)} \sigma^{s,\alpha} \leqslant 1 \qquad \text{randomised scheduler}$$

$$\forall s, \alpha: \quad \sigma^{s,\alpha} \leqslant 1$$

$$\forall s, \alpha: \quad \sum_{t \in S} \overline{\mathcal{P}}(s, \alpha, t) \leqslant 1 \qquad \text{probabilistic choice}$$

$$\forall s, t, \alpha: \quad \overline{\mathcal{P}}(s, \alpha, t) \leqslant 1$$

GP for Two Objectives

Objectives: reach T with probability $\leqslant p$, expected cost to reach G $\leqslant c$

Subject to:

$$\frac{p_{s_I}}{p} \leqslant 1 \qquad \text{reachability}$$

$$\frac{c_{s_I}}{c} \leqslant 1 \qquad \text{expected reward}$$

$$\forall s: \quad \sum_{\alpha \in Act(s)} \sigma^{s,\alpha} \leqslant 1 \qquad \text{randomised scheduler}$$

$$\forall s, \alpha: \quad \sigma^{s,\alpha} \leqslant 1$$

$$\forall s, \alpha: \quad \sum_{t \in S} \overline{\mathcal{P}}(s, \alpha, t) \leqslant 1 \qquad \text{probabilistic choice}$$

$$\forall s, t, \alpha: \quad \overline{\mathcal{P}}(s, \alpha, t) \leqslant 1$$

$$\forall s \in T: \quad p_s = 1 \qquad \text{reach prob of } T$$

$$\forall s \notin T: \quad \frac{\sum_\alpha \sigma^{s,\alpha} \cdot \sum_{t \in S} \overline{\mathcal{P}}(s, \alpha, t) \cdot p_t}{p_s} \leqslant 1 \qquad \text{transition probabilities}$$

GP for Two Objectives

Objectives: reach T with probability $\leqslant p$, expected cost to reach $G \leqslant c$

Subject to:

$$\frac{p_{s_I}}{p} \leqslant 1 \qquad \text{reachability}$$

$$\frac{c_{s_I}}{c} \leqslant 1 \qquad \text{expected reward}$$

$$\forall s: \quad \sum_{\alpha \in Act(s)} \sigma^{s,\alpha} \leqslant 1 \qquad \text{randomised scheduler}$$

$$\forall s, \alpha: \quad \sigma^{s,\alpha} \leqslant 1$$

$$\forall s, \alpha: \quad \sum_{t \in S} \overline{\mathcal{P}}(s, \alpha, t) \leqslant 1 \qquad \text{probabilistic choice}$$

$$\forall s, t, \alpha: \quad \overline{\mathcal{P}}(s, \alpha, t) \leqslant 1$$

$$\forall s \in T: \quad p_s = 1 \qquad \text{reach prob of } T$$

$$\forall s \notin T: \quad \frac{\sum_\alpha \sigma^{s,\alpha} \cdot \sum_{t \in S} \overline{\mathcal{P}}(s, \alpha, t) \cdot p_t}{p_s} \leqslant 1 \qquad \text{transition probabilities}$$

$$\forall s \notin G: \quad \frac{\sum_\alpha \sigma^{s,\alpha} \cdot \left(c(s, \alpha) + \sum_{t \in S} \overline{\mathcal{P}}(s, \alpha, t) \cdot c_t \right)}{c_s} \leqslant 1 \qquad \text{expected costs}$$

Correctness

Use the objective function F now[4]

$$\text{Minimise} \quad \sum_{p \in V} \frac{1}{p} + \sum_{p \in L} \frac{1}{\overline{p}} + \sum_{s,\alpha} \frac{1}{\sigma^{s,\alpha}}$$

yields that all variables p, $\overline{p}$ and $\sigma^{s,\alpha}$ are maximised.

Theorem: The GP with objective function F yields a feasible solution.

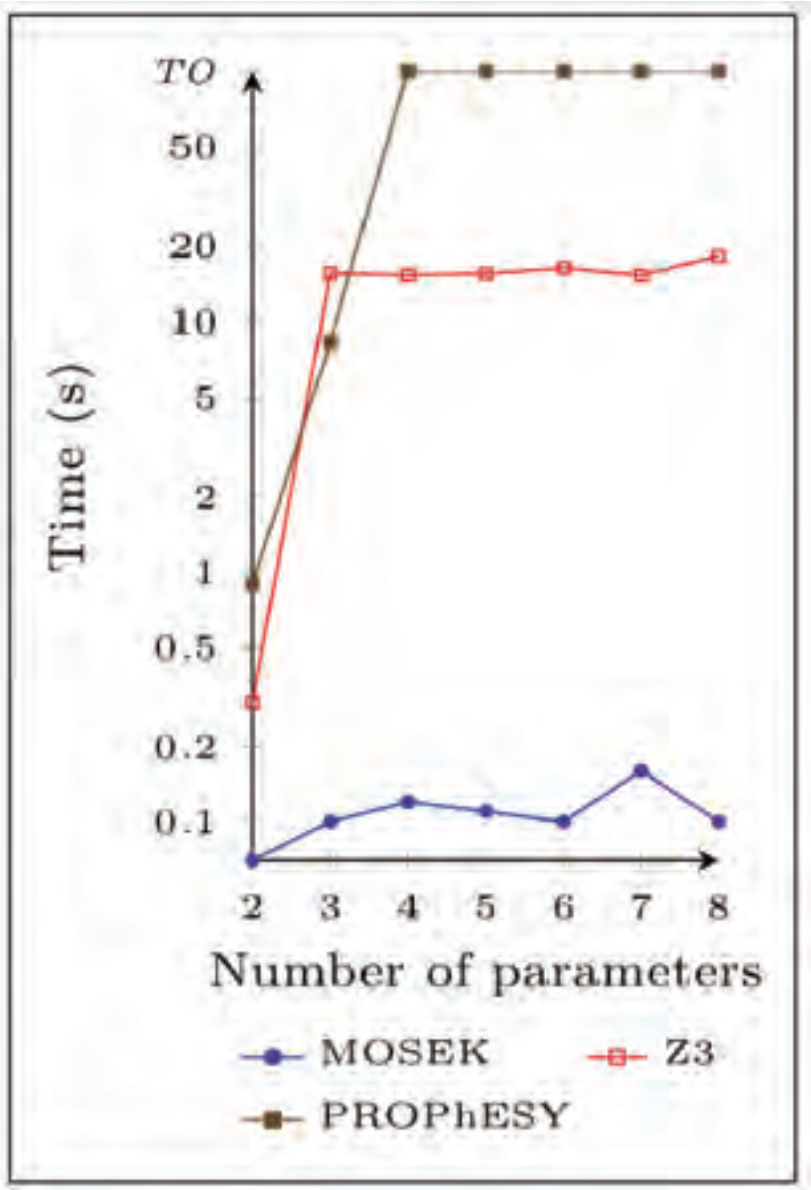
Solving this GP can be done in polynomial time.

[4] Note: the original objective function f is dropped.

Experimental Results

Benchmark	#states	#par	specs	MOSEK (s)		Z3
BRP (pMC)	5382	2	EC, $\mathbb{P}$, $*$	23.17	(6.48)	–
	112646	2	EC, $\mathbb{P}$, $*$	3541.59	(463.74)	–
	112646	4	EC, $\mathbb{P}$, $*$	4173.33	(568.79)	–
	5382	2	EC, $\mathbb{P}$	3.61		904.11
	112646	2	EC, $\mathbb{P}$	479.08		*TO*
NAND (pMC)	4122	2	EC, $\mathbb{P}$, $*$	14.67	(2.51)	–
	35122	2	EC, $\mathbb{P}$, $*$	1182.41	(95.19)	–
	4122	2	EC, $\mathbb{P}$	1.25		1.14
	35122	2	EC, $\mathbb{P}$	106.40		11.49
BRP (pMDP)	5466	2	EC, $\mathbb{P}$, $*$	31.04	(8.11)	–
	112846	2	EC, $\mathbb{P}$, $*$	4319.16	(512.20)	–
	5466	2	EC, $\mathbb{P}$	4.93		1174.20
	112846	2	EC, $\mathbb{P}$	711.50		*TO*
CONS (pMDP)	4112	2	EC, $\mathbb{P}$, $*$	102.93	(1.14)	–
	65552	2	EC, $\mathbb{P}$, $*$	*TO*		–
	4112	2	EC, $\mathbb{P}$	6.13		*TO*
	65552	2	EC, $\mathbb{P}$	1361.96		*TO*

Experimental Results



Epilogue

SMT-based approach:

- Exact
- Requires rational functions
- Fickle SMT performance
- $\approx 10^6$ states, 2 parameters
- Restricted to Markov chains
- CEGAR-like refinement

Epilogue

SMT-based approach:

- Exact
- Requires rational functions
- Fickle SMT performance
- $\approx 10^6$ states, 2 parameters
- Restricted to Markov chains
- CEGAR-like refinement

Parameter lifting approach:

- Approximative
- Off-the-shelf model checking
- No SMT, no rational functions
- $\approx 10^7$ states, 4–5 parameters
- Applicable to MDPs and games
- CEGAR-like refinement

Epilogue

SMT-based approach:

- Exact
- Requires rational functions
- Fickle SMT performance
- $\approx 10^6$ states, 2 parameters
- Restricted to Markov chains
- CEGAR-like refinement

Geometric programming approach:

- Numerical approximation
- Multiple objectives
- $\approx 10^5$ states, 10 parameters
- Applicable to MDPs
- Possibility of richer objectives

Epilogue

SMT-based approach:

- Exact
- Requires rational functions
- Fickle SMT performance
- $\approx 10^6$ states, 2 parameters
- Restricted to Markov chains
- CEGAR-like refinement

Geometric programming approach:

- Numerical approximation
- Multiple objectives
- $\approx 10^5$ states, 10 parameters
- Applicable to MDPs
- Possibility of richer objectives

Significant progress in the last couple of years.

Epilogue

SMT-based approach:

- Exact
- Requires rational functions
- Fickle SMT performance
- $\approx 10^6$ states, 2 parameters
- Restricted to Markov chains
- CEGAR-like refinement

Geometric programming approach:

- Numerical approximation
- Multiple objectives
- $\approx 10^5$ states, 10 parameters
- Applicable to MDPs
- Possibility of richer objectives

Significant progress in the last couple of years.

More info: QEST'14, CAV'15, ATVA'16, TACAS'17 and

`http://moves.rwth-aachen.de/research/tools/prophesy/`