

Simulation-Based Model Checking for Non-Deterministic Systems and Rare Events

12/20/2012 AFOSR Project Review

Prof. Ed Clarke
CMU

emc@cmu.edu
(412) 268-2628



Carnegie Mellon

Dr. David J. Musliner
SIFT

david.musliner@musliner.com
(612) 325-9314



Smart Information Flow Technologies (SIFT)

- 30-person Minneapolis R&D company for 13 years, now \$6M/yr.
- Research emphasis on AI, formal methods, and human-centered systems.
- Multi-agent planning and execution, formal verification, and high-confidence real-time systems for robotics, UAVs, cyber security.

Pilot's Associate



Cyber Security



Abnormal Situation Management



Driver Adaptive Warning System



Agile Information Control Environment



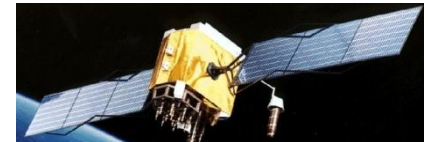
Dynamic Interaction Generation



Playbook UIs



Spacecraft Autonomy



Outline

- Project overview.
- Statistical model checking (background).
- Handling rare events.
- Handling non-determinism.

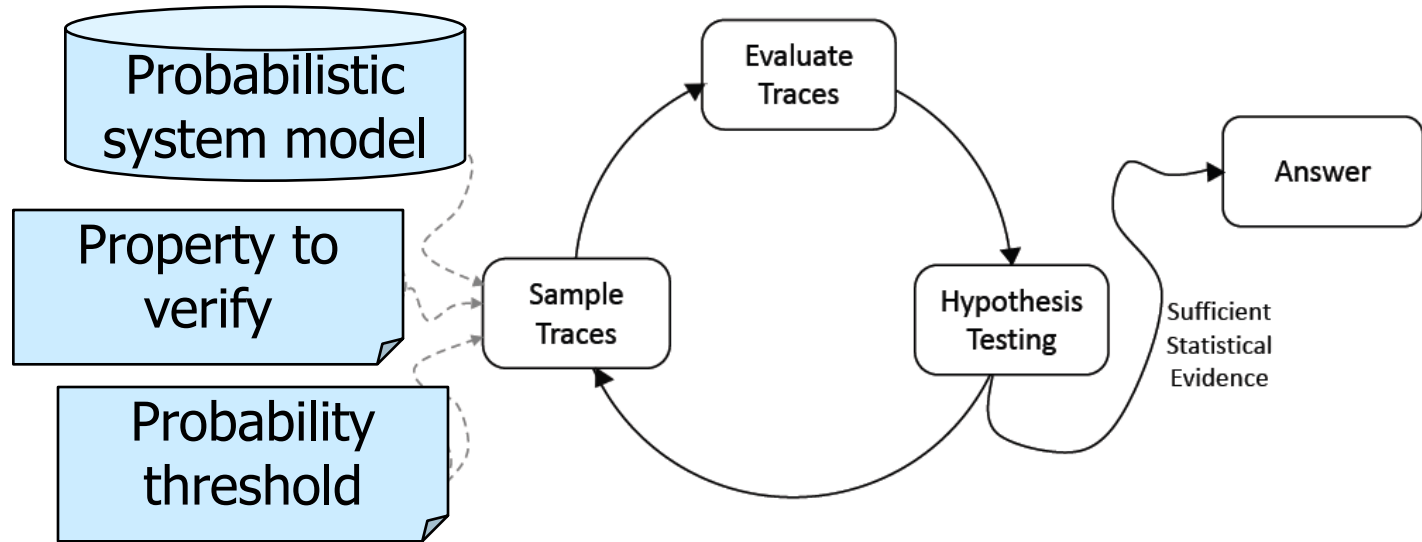
Cyber-Physical System Verification Problem

- Heterogeneous, cyber-physical designs.
- Too uncertain, large, complex, and diverse to be completely or perfectly verified.
 - State spaces may be simply too large.
 - System behavior may be inherently uncertain/stochastic.
- Main objective: *probabilistic* verification of system designs.

Probabilistic Model Checking

- Reasoning about probabilistic models:
 - Discrete-time Markov chains.
 - Continuous-time Markov chains.
 - Markov decision processes.
 - Probabilistic timed automata.
 - More complex systems with no analytic properties.
- Estimation or verification of probabilistic properties; e.g.:
 - “Probability of system failure within 4 hours?”
 - “Expected size of message queue after 30 minutes?”
 - “With probability at least 0.99, the system elects a leader within 1 second.”

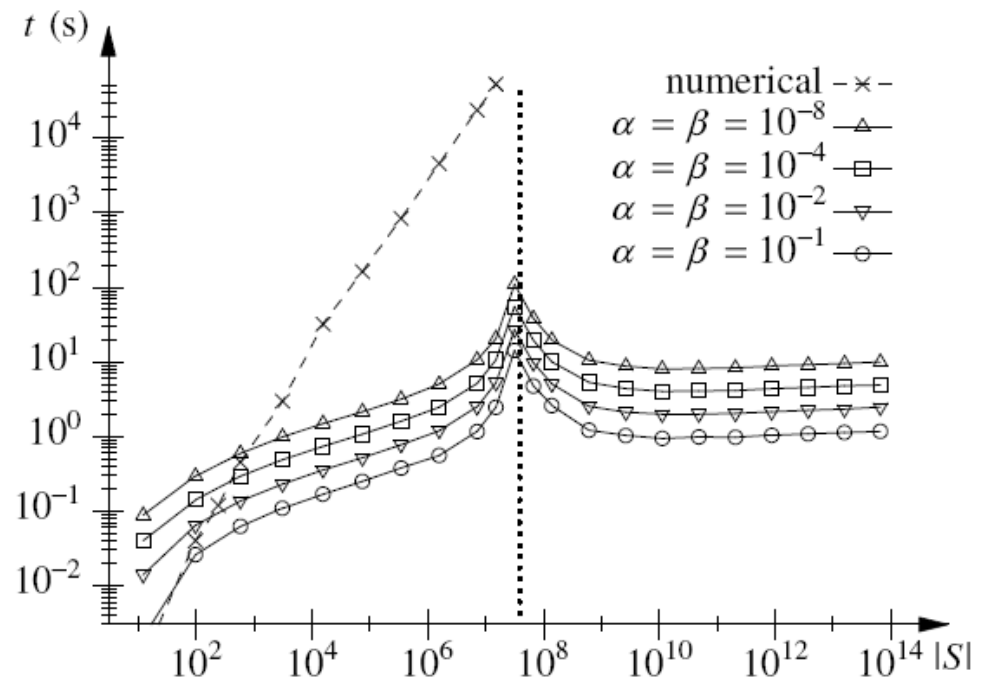
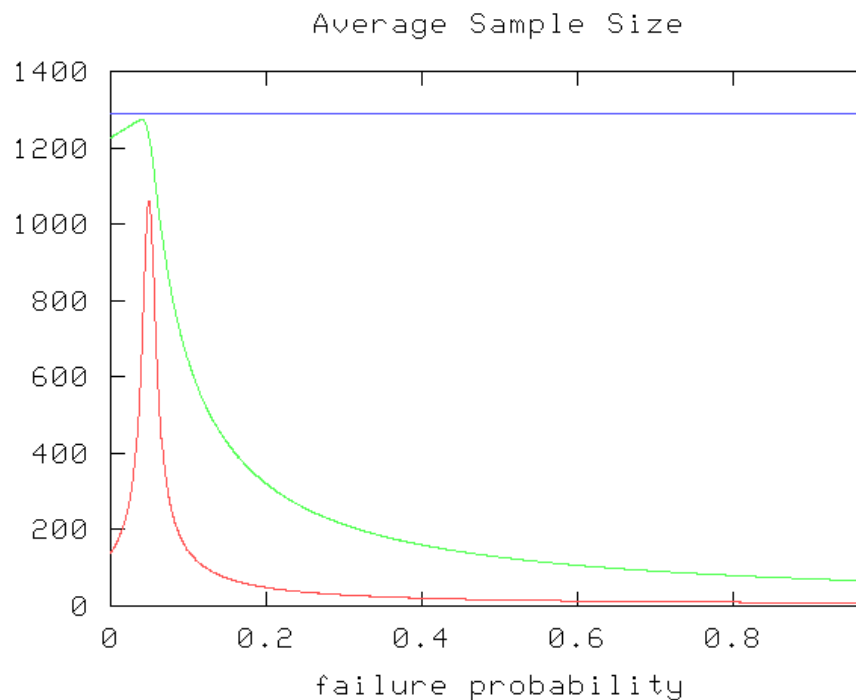
Statistical Hypothesis Testing



- Sample one trace of the system execution.
- Assess whether the execution trace satisfies the property.
- Use statistical methods to decide whether the accumulated samples justify a sufficiently-confident conclusion about the hypothesis.
- If a conclusion is not yet possible, draw more samples.

Statistical Hypothesis Testing Benefits

- Does not require explicit system model or well-formed probability distributions; only a simulation.
- Can scale better than analytic/numeric methods:
 - Analytic/numeric model checking is sensitive to model size.
 - Statistical hypothesis testing is sensitive to horizon and “difficulty” of the question you are asking:



CMU and SIFT have collaborated to develop:

- New science:
 - Statistical model checking methods.
 - Improved termination conditions for statistical hypothesis testing.
- New implementations:
 - Matlab-based tools (CMU).
 - Extended PRISM \rightarrow PRISMATIC (SIFT + CMU).
 - Enhanced CIRCA automatic controller synthesis (SIFT).

This Project

- Handling rare events.
- Handling unquantified non-determinism.
- In Matlab:
 - Arbitrary Stateflow/Simulink models, no analytics possible.
- In PRISMATIC:
 - Structured probabilistic models.
 - Analytic methods applicable.
- In CIRCA:
 - Controller synthesis framework.
 - Complex GSMDP models with no analytics possible.
 - Verification used to assess quality of synthesized controller and guide revisions if it fails.



PRISM: Probabilistic Model Checker

- Supports numeric and sampling techniques:
 - Numeric: Compute results by symbolic state space representation and propagation of distributions.
 - Sampling: Estimate probabilities by averaging a large number of random executions.
 - Useful on very large or complex-form models when numeric model checking is infeasible.
- PRISM is open source (GPL) and easily available.
 - Actively developed by Oxford (Kwiatkowska) & Birmingham (Parker).
 - Supports several input model and property languages.
 - Widely used for teaching/research.
 - Includes many real-world case studies.

PRISMATIC

- Builds on PRISM's existing probabilistic verification capabilities.
- Includes:
 - Statistical hypothesis-testing verification methods.
 - Compositional / Assume-Guarantee Reasoning.
 - Counterexample generation.
- Non-GPL licensing for government use.

CIRCA Controller Synthesis

Available
actions

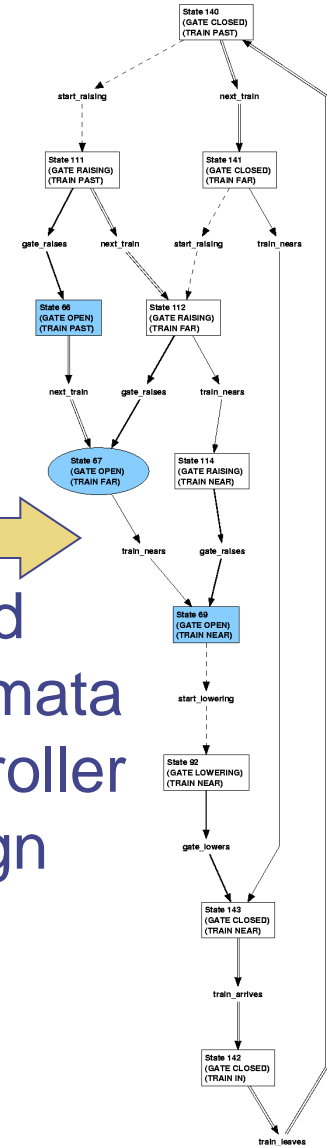
“Non-volitional”
transitions

Goal state
description

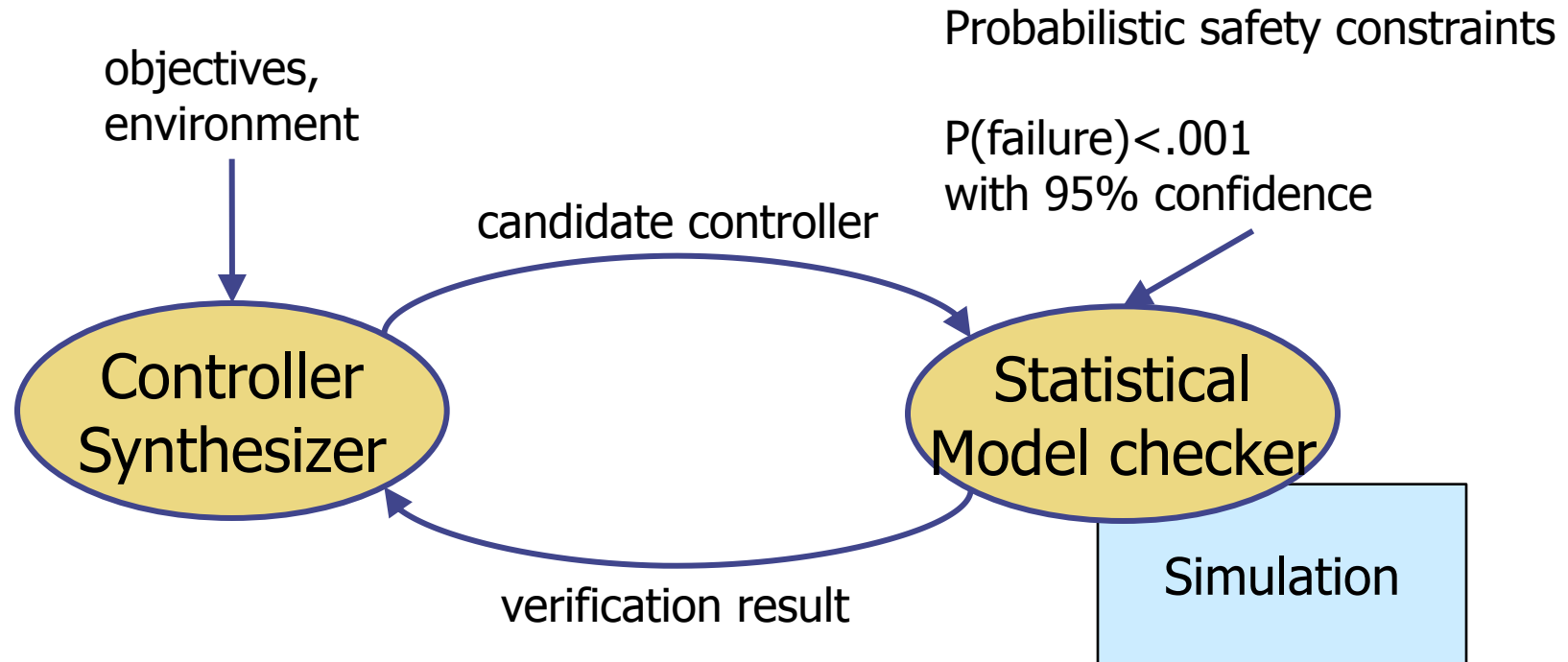
Initial state(s)
description

Controller
Synthesis
Module

Timed
Automata
Controller
Design



Controller Synthesis with Statistical Model Checking



Challenge 1: Rare Events

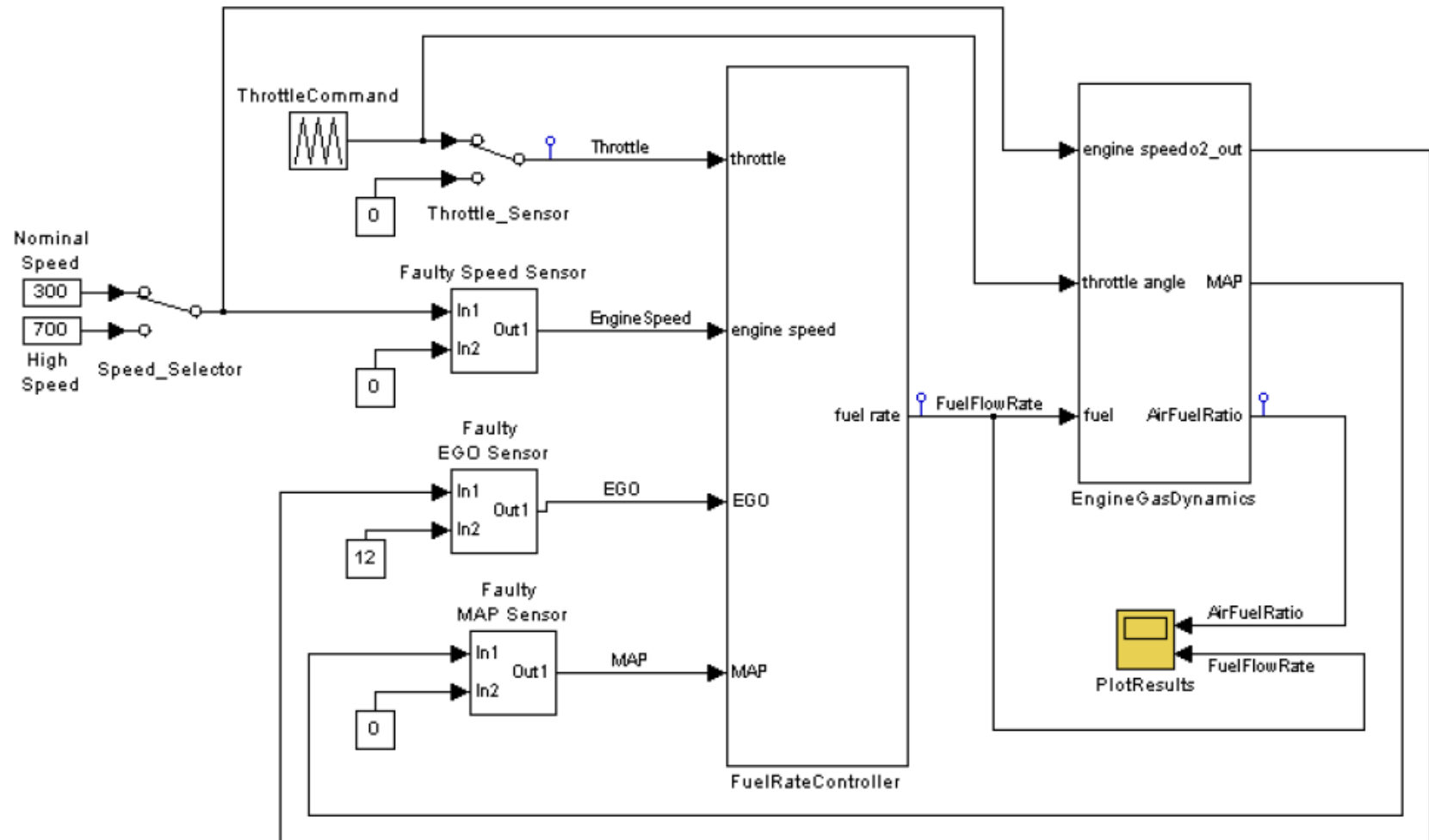
- The number of simulations (coin flips) needed to estimate p accurately grows too large.
 - $p_t = 10^{-9}$ and $\delta = 10^{-2}$ (ie, 1% relative accuracy) we need about 10^{13} samples!!
 - Bayesian estimation requires about 10^6 samples with $p_t=10^{-4}$ and $\delta = 10^{-1}$
- ***Importance sampling*** provides a solution:
 - Increase the probability of rare events (“bias”).
 - Draw samples.
 - Assess overall results by re-weighting the samples.
- An optimal bias exists, for which the re-weighted results are accurate, but cannot be computed directly.

Learning to Improve Biasing Density

- The Kullback-Leibler divergence (cross-entropy) is a measure of “distance” between two densities.
- First used for rare event simulation by Rubinstein (1997).
- In our cross-entropy approach, we use importance sampling to estimate the deviation between an initial bias and optimal bias.
- Choose a first-guess biasing density.
- Run some samples.
- Adjust biasing density and run again.

Example: Fuel Control System

The Stateflow/Simulink model



Verification

- We want to estimate the probability that

$$\mathcal{M}, \text{FaultRate} \models F^{100} G^1(\text{FuelFlowRate} = 0)$$

- “It is the case that within 100 seconds, FuelFlowRate is zero for one second.”
- FaultRate=1/3600s for all three sensors.

Importance Sampling Results

- Step 1: use CE method to estimate optimal biasing density
tilting parameters = $\{1/10, 1/10, 1/10\}$
- Step 2: run importance sampling to estimate probability

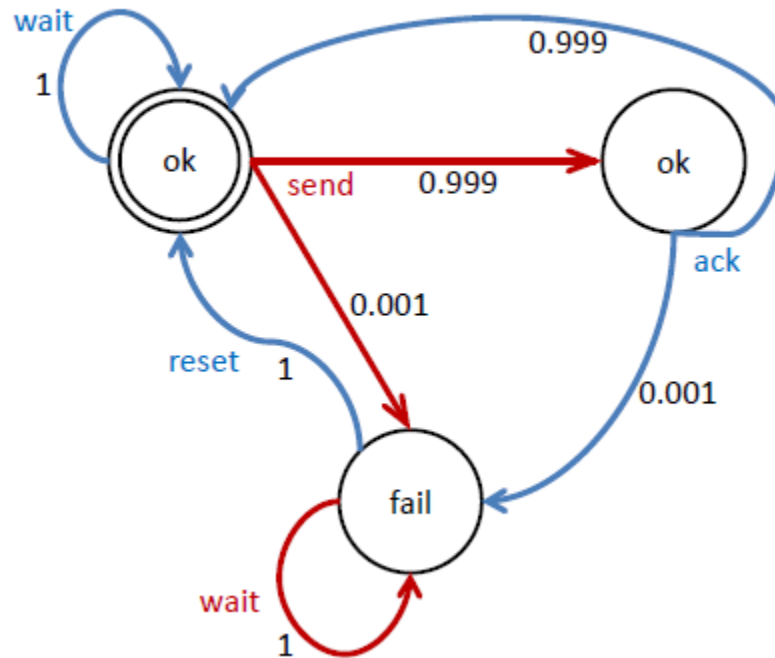
		Estimate	Relative error	Time (h)
Samples	Step 1: 1,000 Step 2: 10,000	5.1×10^{-15}	0.47	1.7
	Step 1: 10,000 Step 2: 100,000	2.17×10^{-14}	0.13	17.8

Importance Sampling Status

- Implemented in Matlab to produce results just shown (HSCC'12 paper).
- Implemented in PRISMATIC and reproduced limited set of test results.
- Implemented in CIRCA, with ongoing development.
- Discovered issue with biasing probability of action outcomes instead of entire plan traces.
 - Wrote and submitted a paper about this problem and techniques to overcome it.
 - Implemented a graph compilation technique to avoid this in CIRCA.

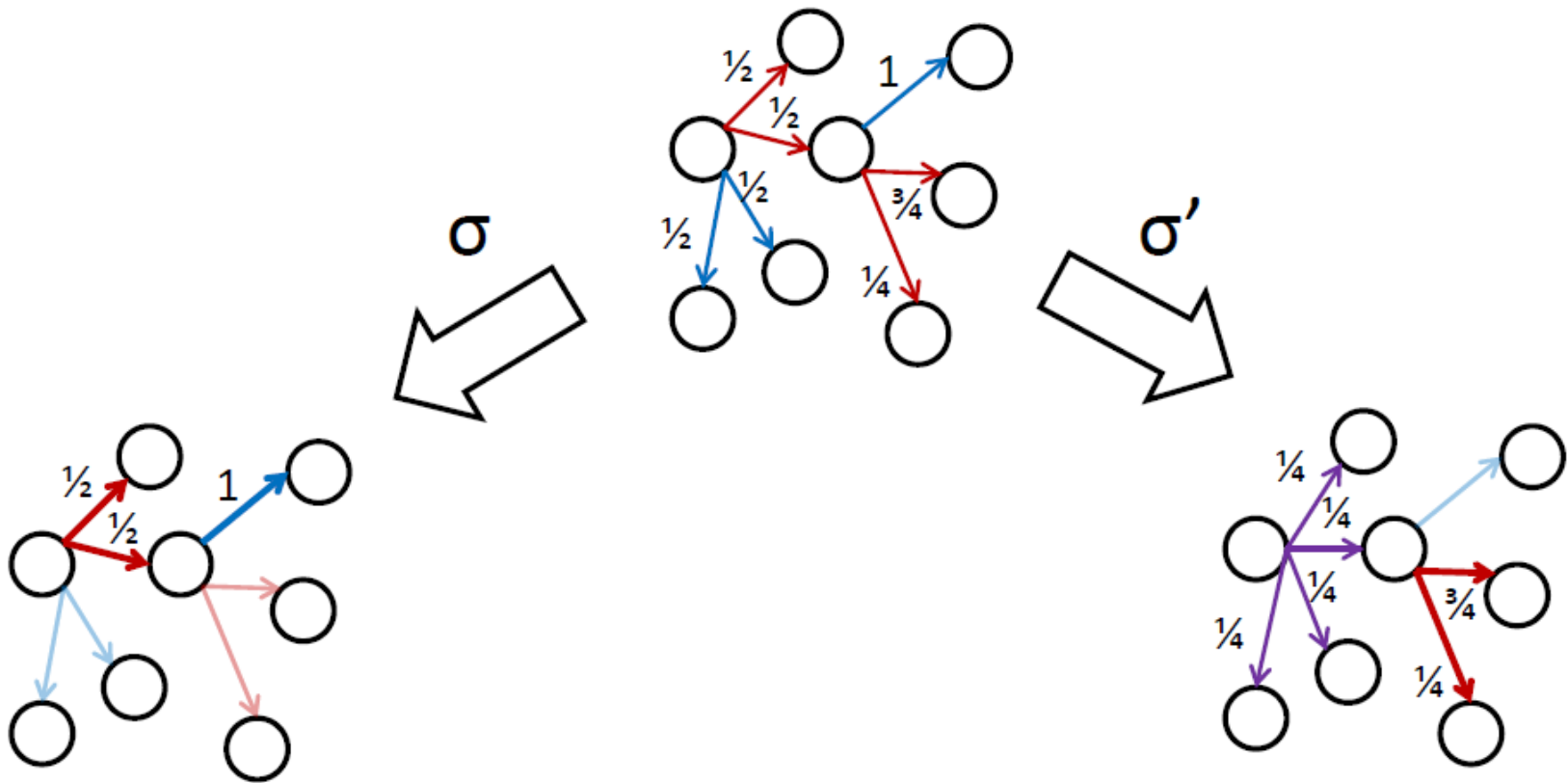
Challenge 2: Non-Determinism

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



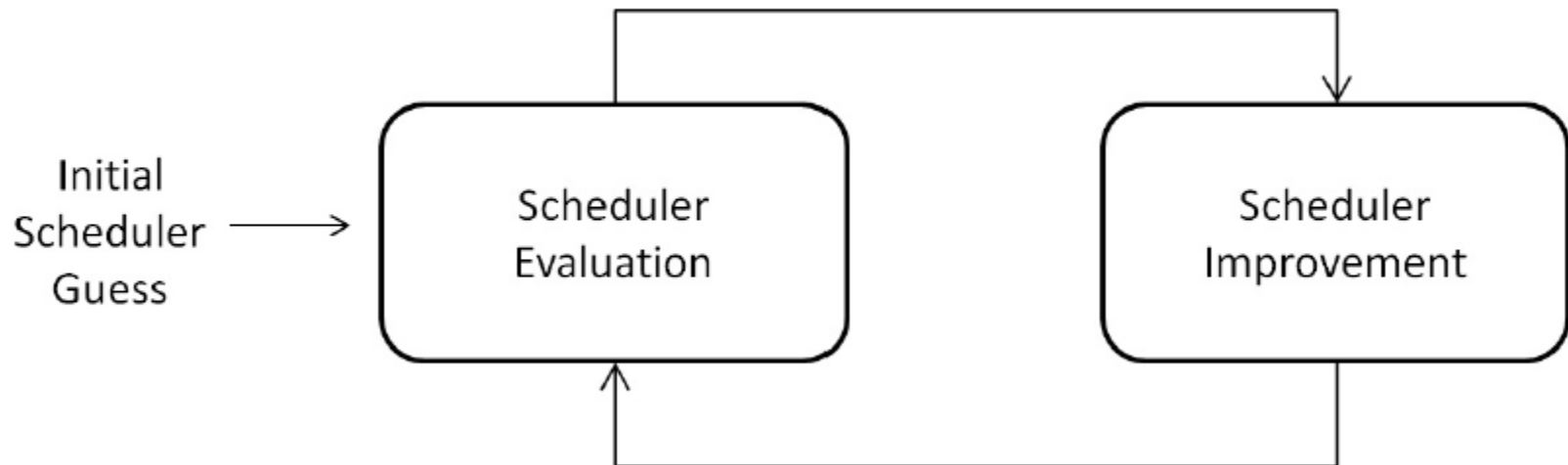
Resolving Non-Determinism

- Memoryless stochastic policy or “scheduler” can resolve non-determinism.
- Specifies choices in each state:



Approach

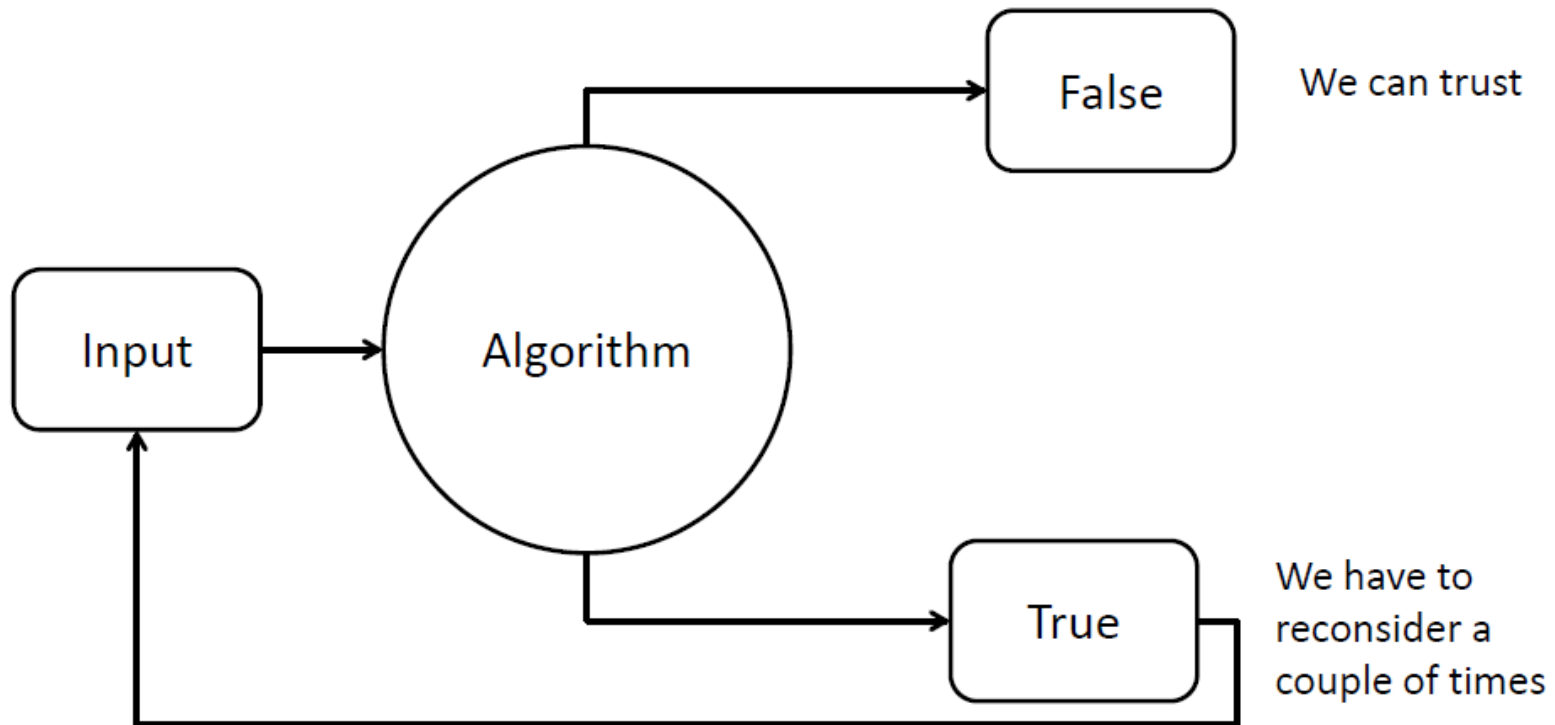
- Learn the most adversarial choices at each state, by successively refining an initial guess.



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

Restarts Improve Confidence

Algorithms like this are called “False-biased Monte Carlo Algorithms”



Confidence increases exponentially with the number of times we restart.

Results on Highly Structured Models (QEST'12 paper)

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

Takeaways

- Symmetry makes the number of “meaningful” actions relatively small;
- SMC works well in highly structured systems;
- Exact methods still work best in most cases;

Results on Moderately Structured Model

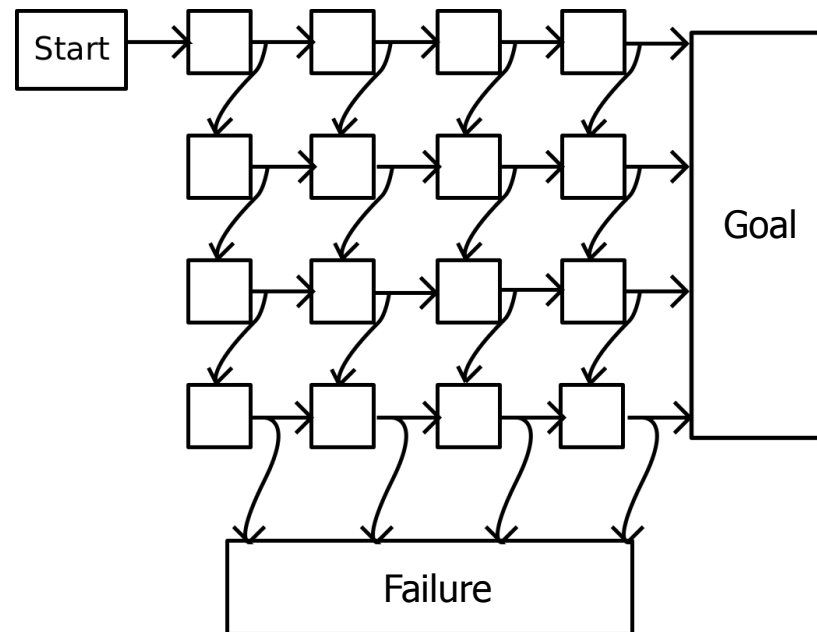
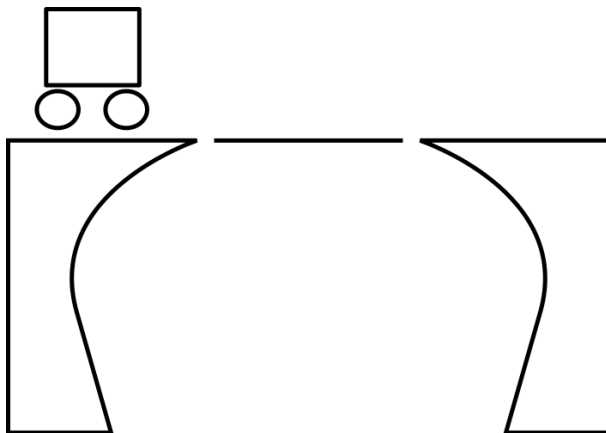
Robot $n = 50$ $r = 1$	θ	0.9	0.95	0.99	PRISM
	out	F	F	F	0.999
	t	23.4	27.5	40.8	1252.7
Robot $n = 50$ $r = 2$	θ	0.9	0.95	0.99	PRISM
	out	F	F	F	0.999
	t	71.7	73.9	250.4	3651.045
Robot $n = 75$ $r = 2$	θ	0.95	0.97	0.99	PRISM
	out	F	F	F	timeout
	t	382.5	377.1	2676.9	timeout
Robot $n = 200$ $r = 3$	θ	0.85	0.9	0.95	PRISM
	out	F	F	T	timeout
	t	903.1	1129.3	2302.8	timeout

Takeaways

- Exact methods cannot exploit symmetry so much;
- Number of really “meaningful” actions still relatively small;
- SMC works very well in structured systems;

Non-Determinism Status

- Implemented in PRISMATIC (QEST'12 paper).
- Started improving CIRCA to properly handle non-determinism in statistical methods.
 - Wrote simple “trap doors” CIRCA domain, with probabilities and non-determinism.
 - Modeled and solved the trap doors domain in PRISM, so we can check our CIRCA results.



Next Steps

- Investigate cross-entropy methods for MDP verification.
- Investigate numerical techniques for variance minimization.
- Finish handling non-determinism in CIRCA.
- Evaluate performance on trap-door and other domains.

The End

- Questions?

Financial Status

- As of Nov 30, CMU has spent \$38.2K of \$141K first-year budget, or 27%.
 - Nonlinear student costs.
- As of Dec 15, SIFT has spent \$129.6K of \$175K first-year budget, or 74%.
 - Essentially on linear plan.

Traditional Model Checking

- Exhaustively explores system state space.
- Formally and conclusively proves operational properties – no ambiguity.
- Does not scale well to some kinds of large systems.
- Does not handle models with uncertainty (e.g., probabilities or unquantified non-determinism).

PRISM System Design Language Compatibility

- PRISM modeling language:
 - Simple but flexible high-level language.
 - Based on guarded commands + finite-ranging variables.
- Other formalisms supported via language-level translation:
 - Probabilistic/stochastic π -calculus.
 - Stochastic Petri nets.
 - MODEST stochastic process algebra.
 - PEPA and Bio-PEPA.
 - SBML (Systems Biology Markup Language).

PRISM Property Specification Language

- Subsumes:
 - PCTL (for DTMCs, MDPs, PTAs).
 - CSL (for CTMCs).
 - LTL, PCTL* (for DTMCs, MDPS, CTMCs).
 - CTL (non-probabilistic; all models) [in progress].
- Also includes:
 - Quantitative extensions for estimation queries.
 - Arithmetic expressions.
 - Cost and reward extensions.