Multi-Objective Parameter Fitting in Parametric Probabilistic Hybrid Automata

From Automatic Verification Using Constraints to Automatic Modeling by Mining Constraints

Martin Fränzle (CvOU Oldenburg)

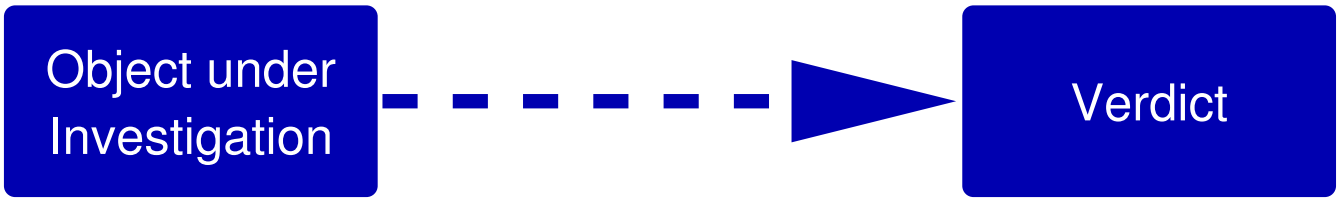
joint work with

Alessandro Abate (Oxford University)
Sebastian Gerwinn (OFFIS e.V., Oldbg.)
Joost-Pieter Katoen (RWTH Aachen)
Paul Kröger (CvOU Oldenburg)

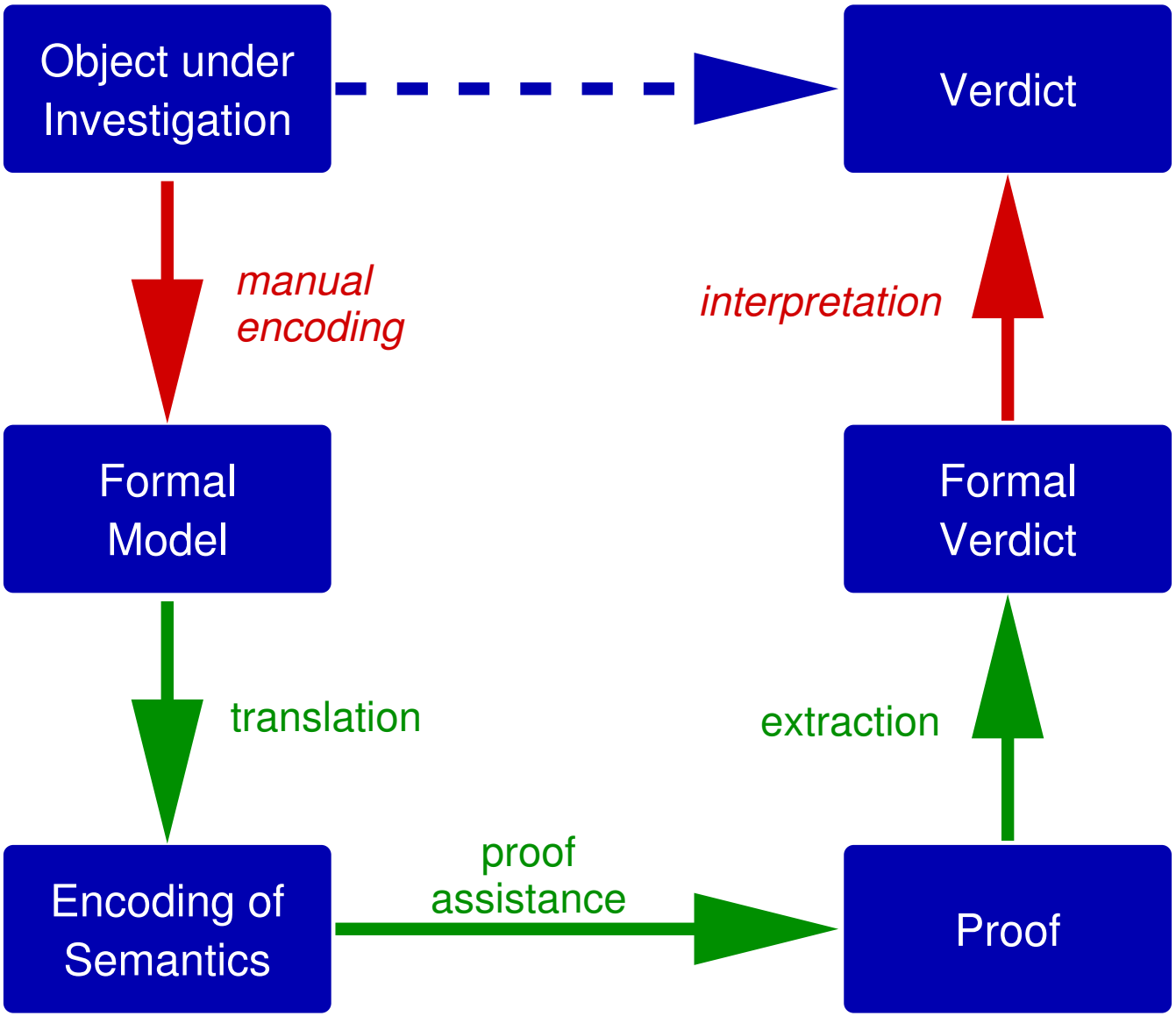
based on technologies contributed by several further AVACS researchers, a.o.

Bernd Becker · Andreas Eggers · Christian Herde
Holger Hermanns · Stefan Kupferschmid · Stefan Ratschan
Karsten Scheibler · Tobias Schubert · Tino Teige

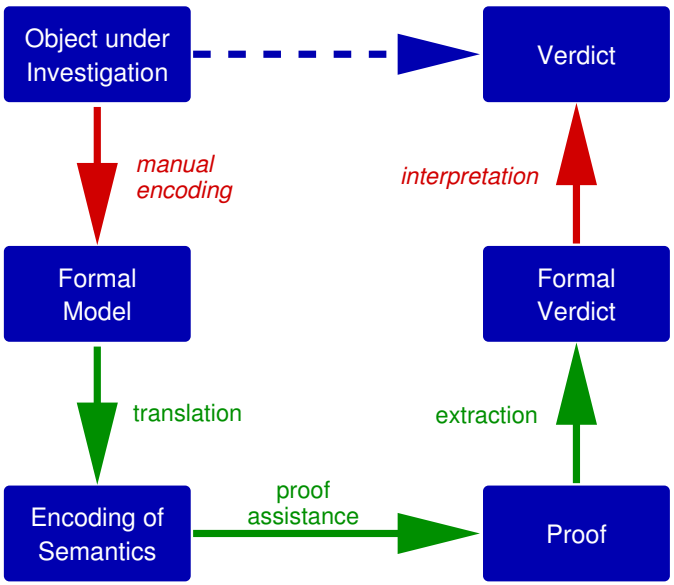
The traditional formal verification cycle



The traditional formal verification cycle



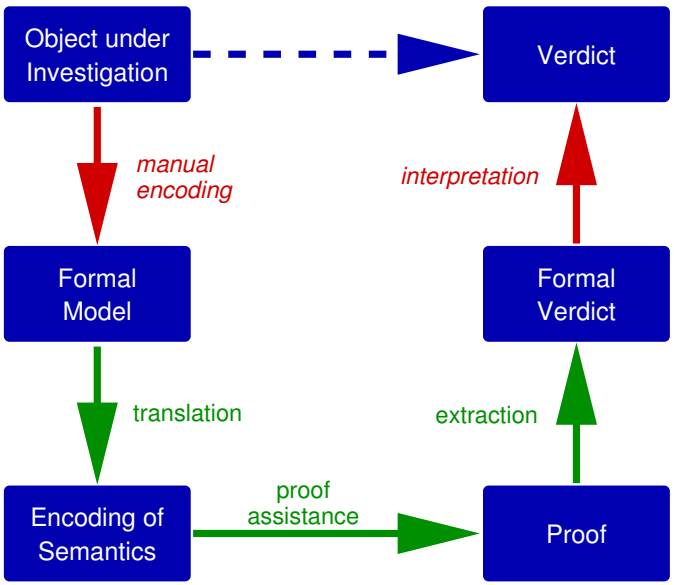
The traditional formal verification cycle



But what if

- faithful formal modeling is too complex to be feasible?
- object under investigation is an embedded system that learns part of its behavior only after deployment?
- we face a long-term autonomous system that may eventually enter unpredictable (i.e., impossible to model a priori) environments & system configurations?

The traditional formal verification cycle



But what if

- faithful formal modeling is too complex to be feasible?
- object under investigation is an embedded system that learns part of its behavior only after deployment?
- we face a long-term autonomous system that may eventually enter unpredictable (i.e., impossible to model a priori) environments & system configurations?

Such applications become increasingly relevant, *challenging our approaches to verification.*

The Mission: Bridge the Gap

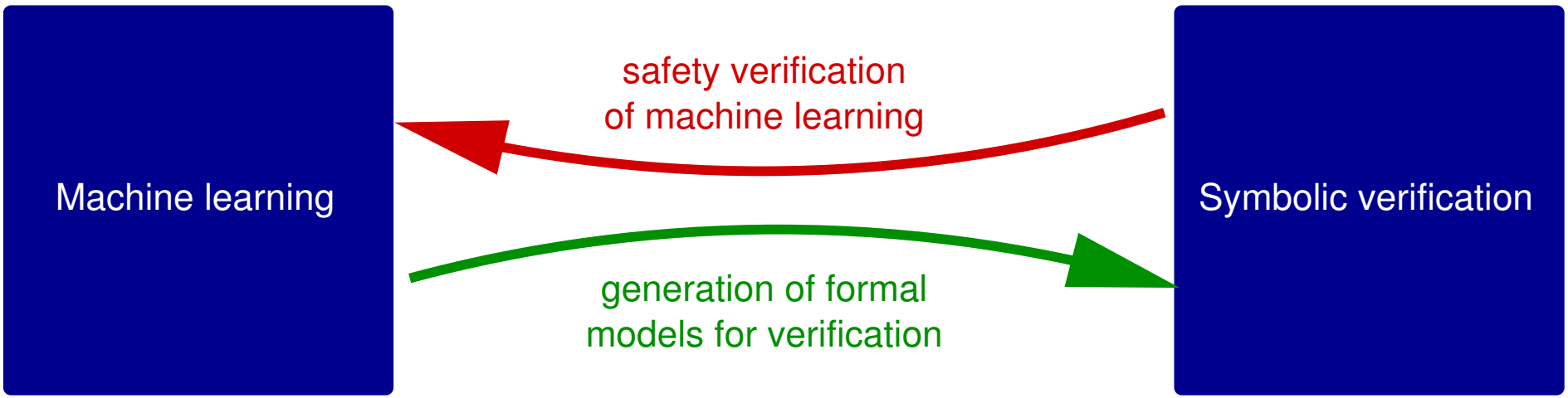
Machine learning

Symbolic verification

The Mission: Bridge the Gap

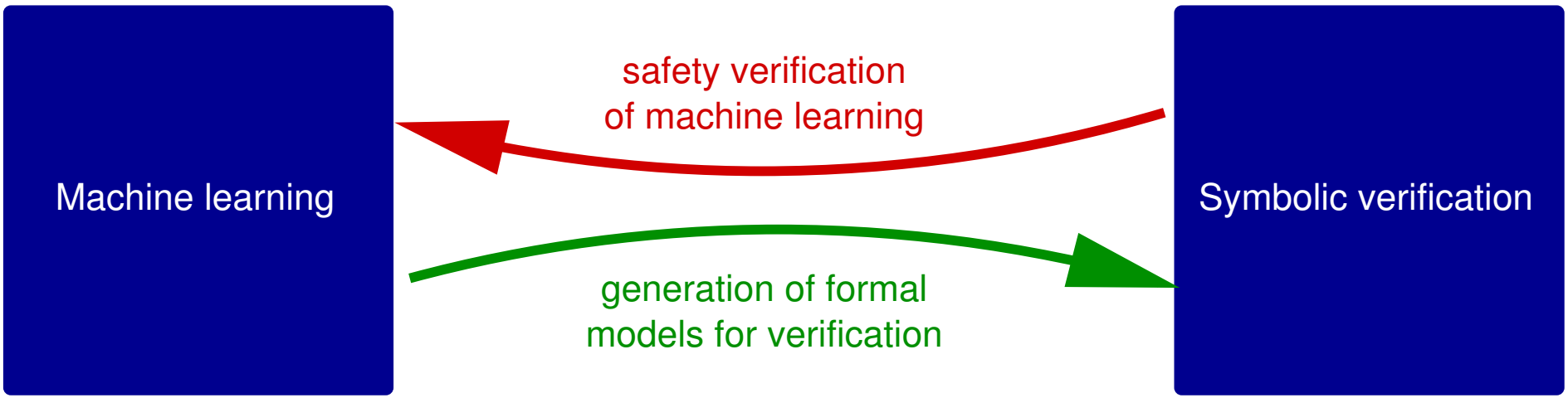
- Need for mechanically supplying safety certificates for ML etc. (static and/or run-time verification)

The Mission: Bridge the Gap



- Need for mechanically supplying safety certificates for ML etc. (static and/or run-time verification)
- May want to exploit ML techniques to bridge the modeling gap
  - when entering unknown / partially known environments, ...
  - when faced with overly complex modeling task.

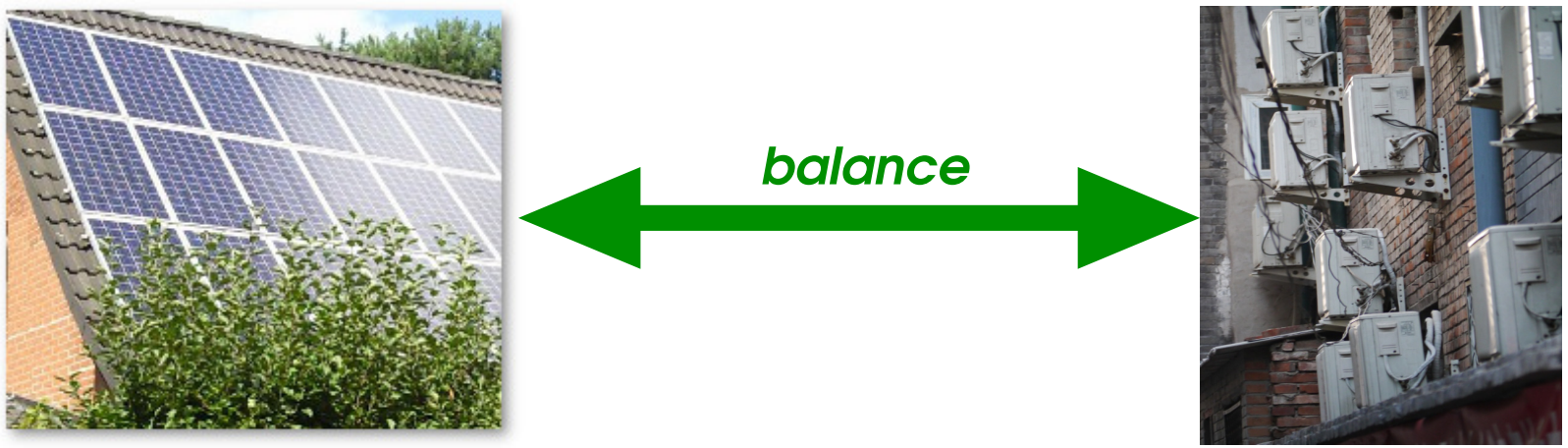
The Mission: Bridge the Gap



- Need for mechanically supplying safety certificates for ML etc. (static and/or run-time verification)
- May want to exploit ML techniques to bridge the modeling gap
  - when entering unknown / partially known environments, ...
  - when faced with overly complex modeling task.

Example: Demand-Response Schemes in Smart Grids

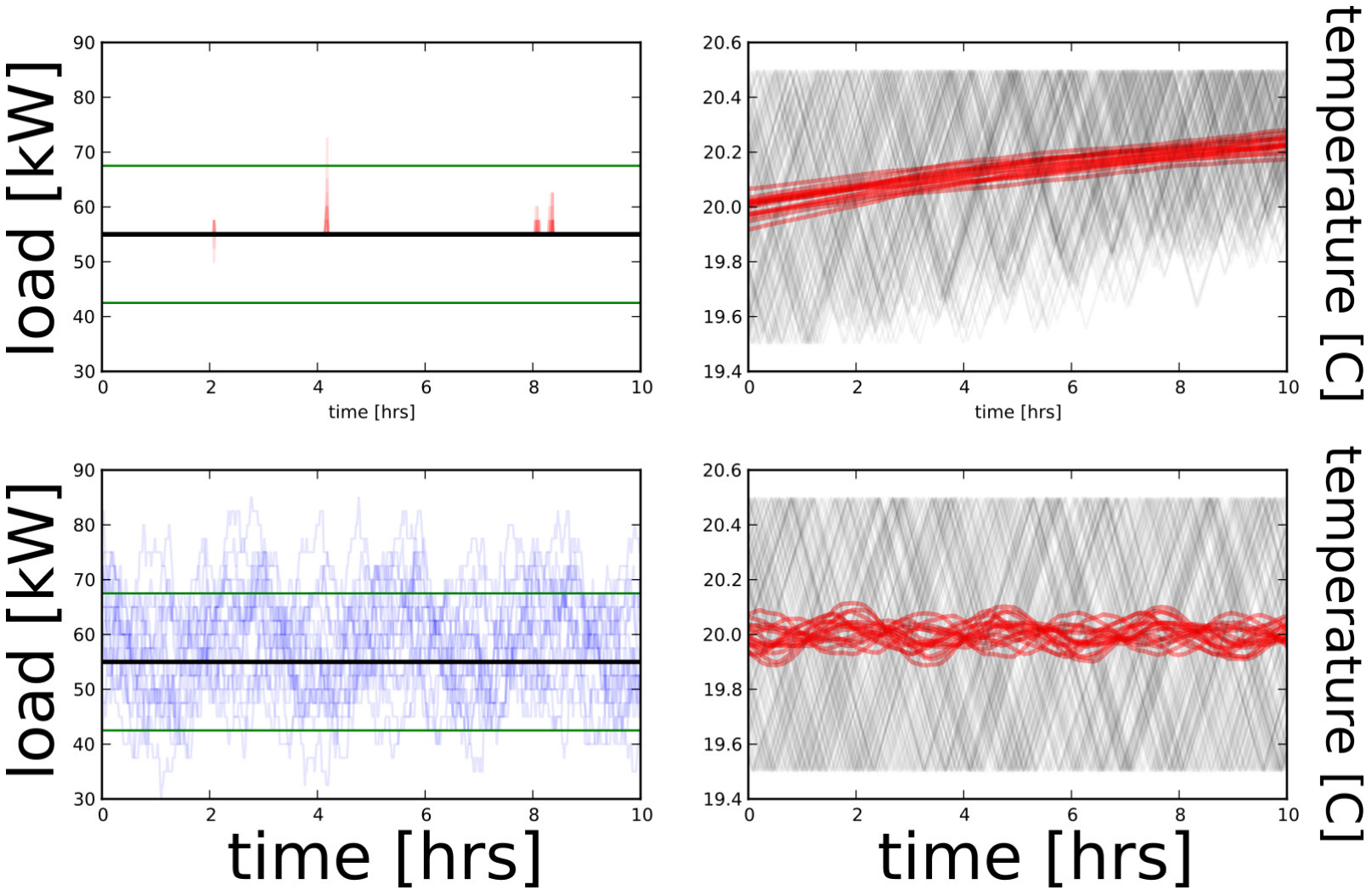
Demand Response: Supplying Reserve Power by Thermostatically Ctrl.ed Loads (TCLs) [Callaway 2009]



Idea: Control power demand by (marginally) modifying switching thresholds of AC systems.

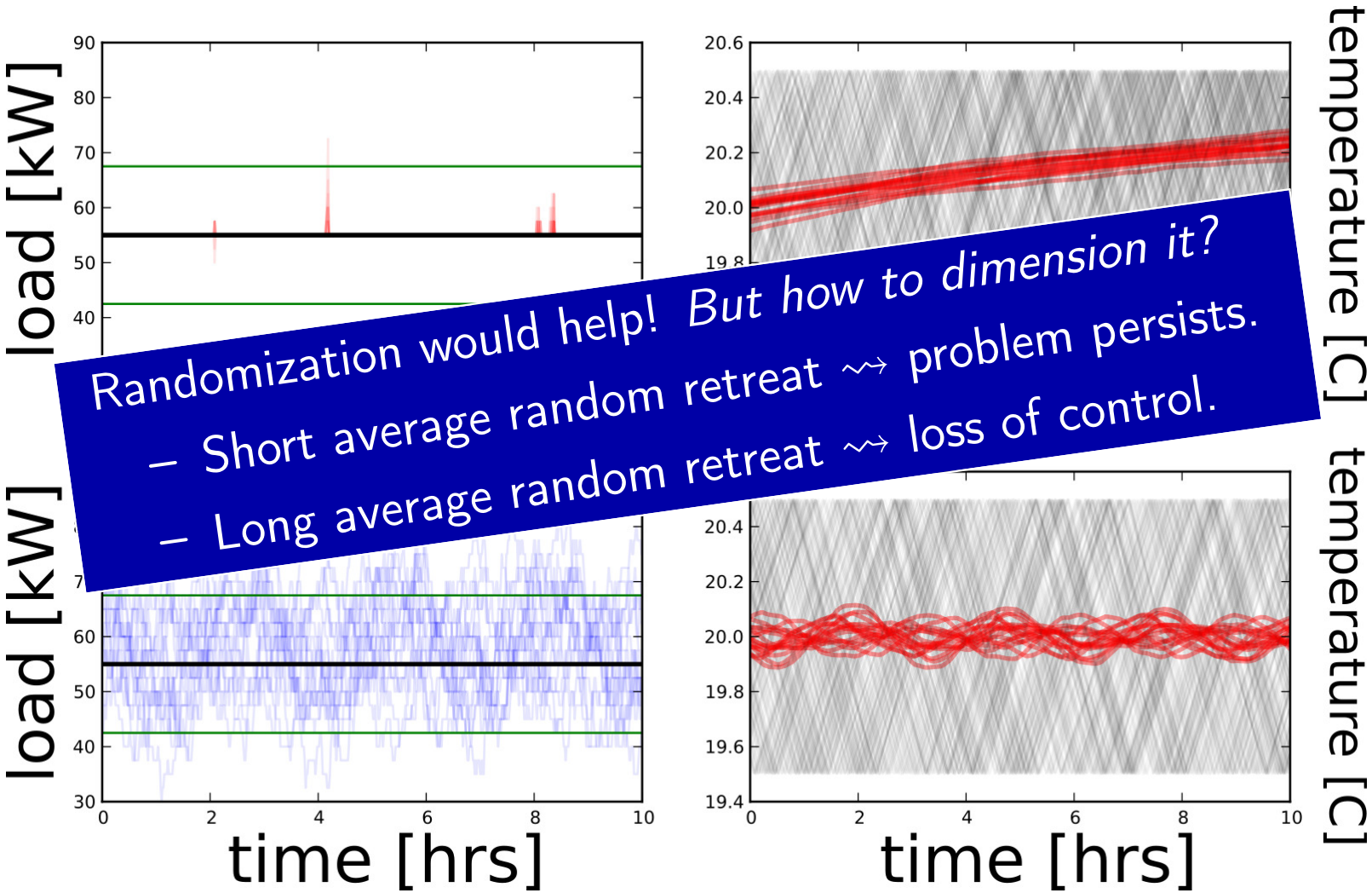
- On power shortage, provide reserve power by switching off early / switching on late.
- On excess power, consume reserve power by switching off late / switching on early.
- Unnoticeable to residents due to marginal adjustments to switching thresholds.

Multiple Similar TCLs ($N = 50$) — Simulation



Externally controlled (power target 55 kW) vs. uncontrolled ensemble.
Control strategy: switch off coldest households if power target exceeded.

Multiple Similar TCLs ($N = 50$) — Simulation

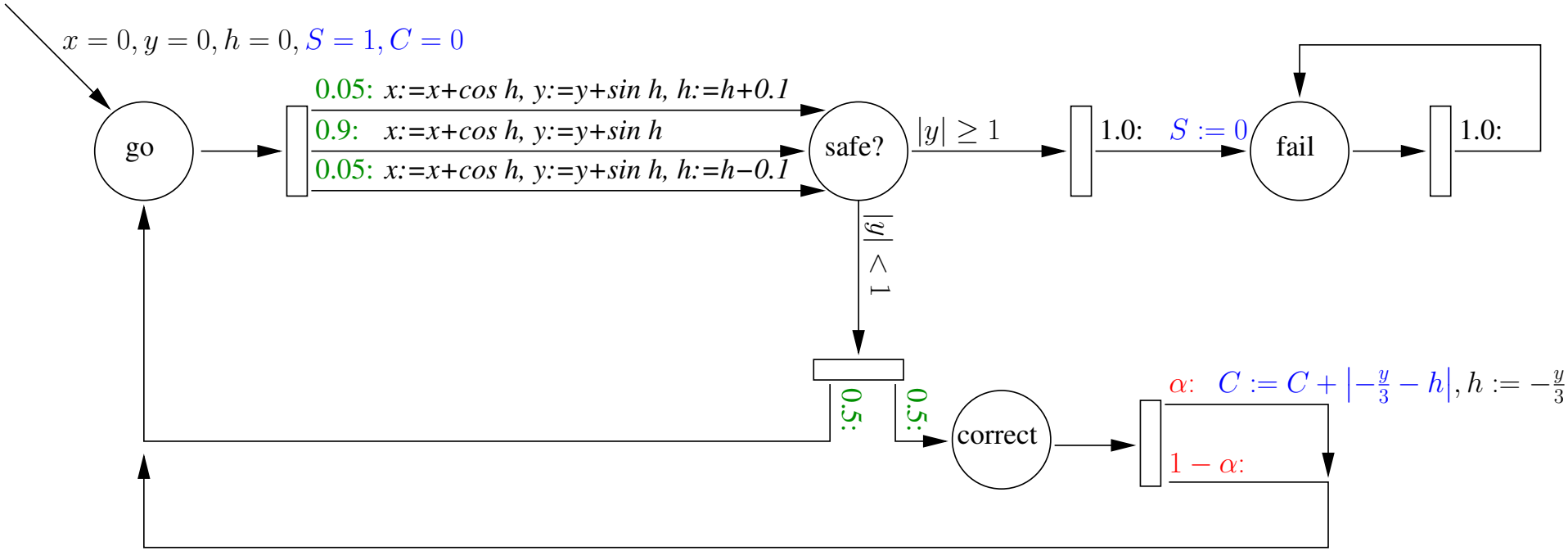


Externally controlled (power target 55 kW) vs. uncontrolled ensemble.
Control strategy: switch off coldest households if power target exceeded.

The Formal Model

Parametric Probabilistic HA

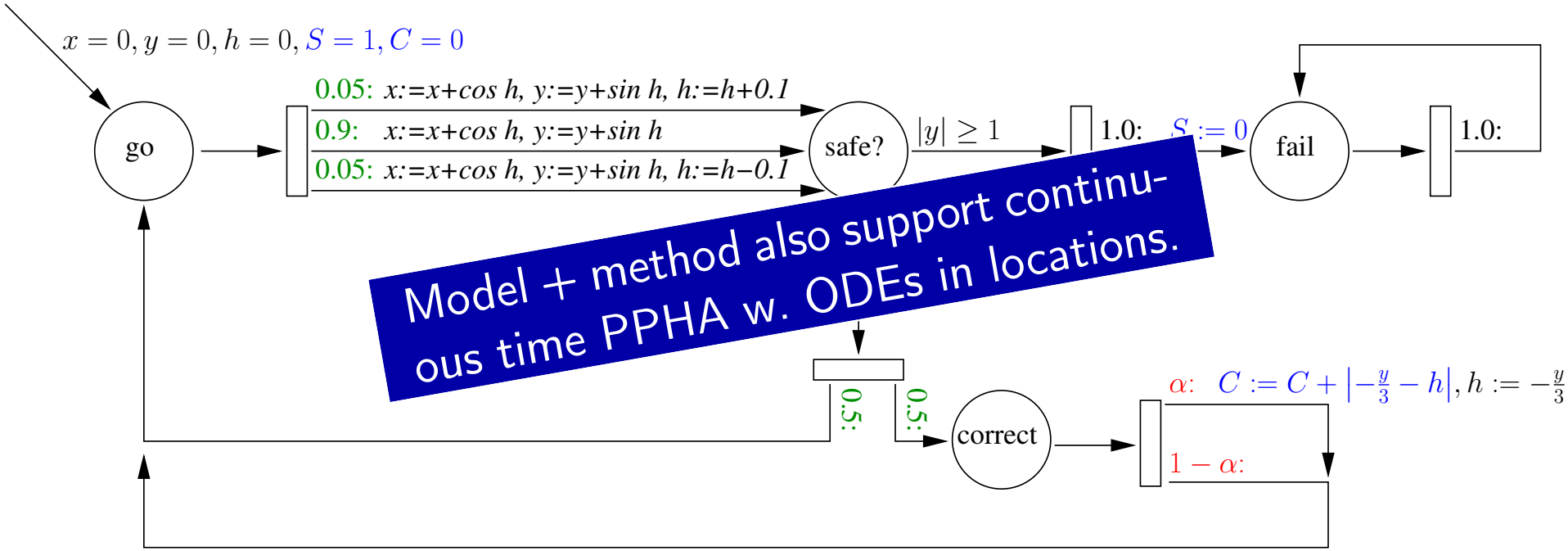
A (discrete time) Parametric Probabilistic HA



Car maneuvre: Keep lane while driving along a road.

- Measurement of position in lane fails with probability 0.5.
- Upon success, do occasional (due to cost associated) corrections of heading angle h by proportional control.
  - Parameter α controls frequency of corrective actions.
- Two reward / cost variables:
  - C records accumulated cost of corrective steering actions,
  - S records successful stay in lane.

A (discrete time) Parametric Probabilistic HA



Car maneuvre: Keep lane while driving along a road.

- Measurement of position in lane fails with probability 0.5.
- Upon success, do occasional (due to cost associated) corrections of heading angle h by proportional control.
  - Parameter α controls frequency of corrective actions.
- Two reward / cost variables:
  - C records accumulated cost of corrective steering actions,
  - S records successful stay in lane.

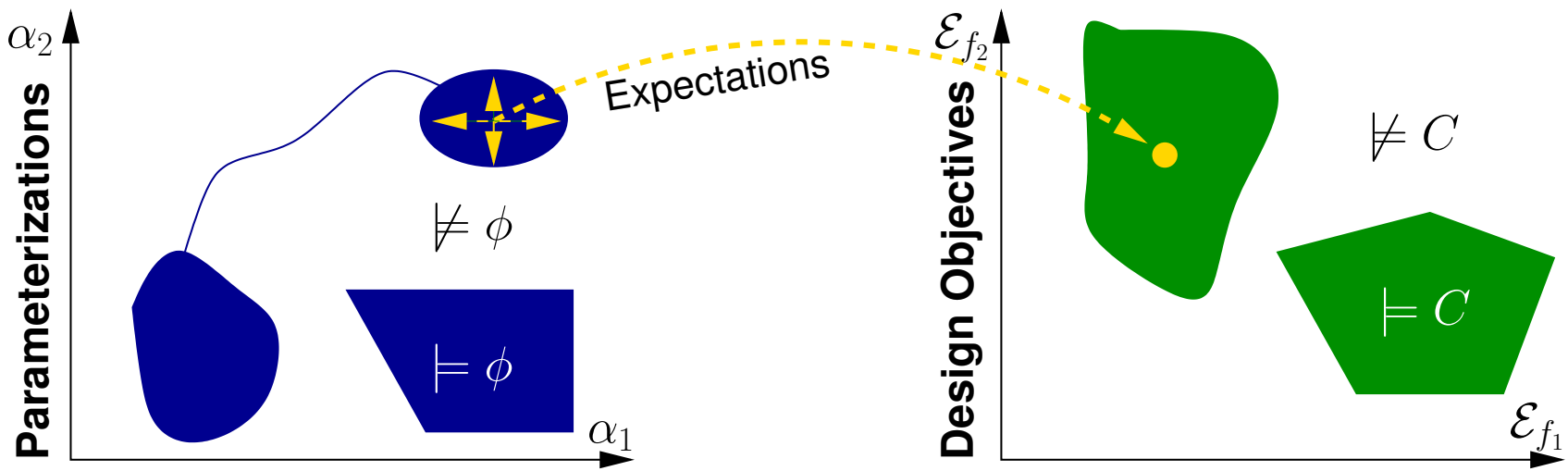
The parameter synthesis problem

Given

1. a PPHA A, featuring
   - a vector $\vec{\alpha} = (\alpha_1, \ldots, \alpha_k)$ of parameters,
   - a vector $\vec{f} = (f_1, \ldots, f_n)$ of reward (or cost) functions,
2. a constraint ϕ over $\vec{\alpha}$ specifying the possible parameter instances, and
3. a constraint C over $\mathcal{E}_{\vec{f}}$ specifying the (multi-objective) design goal,

find (or prove non-existence of) a parameter instance $\theta \in \mathbb{R}^k$ that

1. satisfies ϕ and
2. yields expected rewards $\mathcal{E}[\vec{f}, \theta]$ satisfying C.



Approach

1. Substitution of parametric probabilities in the system model by fixed substitute probabilities;
2. Introduction of counters into the model counting how frequently such substitutes have been chosen along a simulation run;
3. Statistical model checking of the modified model, yielding estimates of the expected costs/rewards in the non-parametric substitute model;
4. Exploitation of the re-normalization equations of importance sampling for obtaining a symbolic expression of the (estimated) parameter dependency of the costs/rewards;
5. Simplification of that expression by means of merging terms;
6. Use of SMT solving over, a.o., higher-order polynomials for determining suitable parameters.

Estimating Expectations by Sampling

Classical sampling

- $p(\cdot;\theta)$ be the parameter-dependent density function of random variable X;
- $\theta^* \models \phi$ be a parameter instance;
- $f : X \to [a, b]$ be a bounded reward function.

Expectation of f depending on θ:

$$\mathcal{E}[f;\theta] = \sum_{x \in X} f(x)p(x;\theta) \tag{1}$$

Classical sampling

- $p(\cdot;\theta)$ be the parameter-dependent density function of random variable X;
- $\theta^* \models \phi$ be a parameter instance;
- $f : X \to [a, b]$ be a bounded reward function.

Expectation of f depending on θ:

$$\mathcal{E}[f;\theta] = \sum_{x \in X} f(x)p(x;\theta) \tag{1}$$

Monte-Carlo approximation of expectation of f in θ^*:

1. Use randomized simulation faithfully representing $p(\cdot, \theta^*)$ to generate n samples $x_1, \ldots, x_m \in X$.
2. Compute the empirical mean

$$\tilde{\mathcal{E}}[f;\theta^*] = \frac{1}{N} \sum_{i=1}^{N} f(x_i) \tag{2}$$

of the sampled f values.

Quality of the estimate

For large numbers of samples N, grossly outlying estimates are unlikely.

Quality of the estimate

For large numbers of samples N, grossly outlying estimates are unlikely.

Hoeffding's inequality [Hoeffding 1963] yields

$$P\left(\tilde{\mathcal{E}}[f;\theta^*] - \mathcal{E}[f;\theta^*] \geq +\varepsilon\right) \leq \exp\left(-2\frac{\varepsilon^2 N}{(b-a)^2}\right) \quad , \tag{3a}$$

$$P\left(\tilde{\mathcal{E}}[f;\theta^*] - \mathcal{E}[f;\theta^*] \leq -\varepsilon\right) \leq \exp\left(-2\frac{\varepsilon^2 N}{(b-a)^2}\right) \quad . \tag{3b}$$

Quality of the estimate

For large numbers of samples N, grossly outlying estimates are unlikely.

Hoeffding's inequality [Hoeffding 1963] yields

$$P\left(\tilde{\mathcal{E}}[f;\theta^*] - \mathcal{E}[f;\theta^*] \geq +\varepsilon\right) \leq \exp\left(-2\frac{\varepsilon^2 N}{(b-a)^2}\right) \quad , \tag{3a}$$

$$P\left(\tilde{\mathcal{E}}[f;\theta^*] - \mathcal{E}[f;\theta^*] \leq -\varepsilon\right) \leq \exp\left(-2\frac{\varepsilon^2 N}{(b-a)^2}\right) \quad . \tag{3b}$$

- Thus, sampling can be used for determining (with confidence) whether a parameterized instance of a PPHA, i.e., a PHA, satisfies the design objective C.
  - Build a formula which determines whether *all* the ε neighbourhood of the empirical mean satisfies C; check by SMT solving.
- The multi-objective parameter fitting problem can then in principle be solved by sampling the parameter space.

Quality of the estimate

For large numbers of samples N, grossly outlying estimates are unlikely.

Hoeffding's inequality [Hoeffding 1963] yields

$$P\left(\tilde{\mathcal{E}}[f;\theta^*] - \mathcal{E}[f;\theta^*] \geq +\varepsilon\right) \leq \exp\left(-2\frac{\varepsilon^2 N}{(b-a)^2}\right) \quad , \tag{3a}$$

$$P\left(\tilde{\mathcal{E}}[f;\theta^*] - \mathcal{E}[f;\theta^*] \leq -\varepsilon\right) \leq \exp\left(-2\frac{\varepsilon^2 N}{(b-a)^2}\right) \quad . \tag{3b}$$

- Thus, sampling can be used for determining (with confidence) whether a parameterized instance of a PPHA, i.e., a PHA, satisfies the design objective C.
  - Build a formula which determines whether *all* the ε neighbourhood of the empirical mean satisfies C; check by SMT solving.
- The multi-objective parameter fitting problem can then in principle be solved by sampling the parameter space.
- But this approach is plagued by the curse of dimensionality; instead need a constructive form of generalizing from samples.

Importance Sampling

The classical, non-symbolic version

Importance sampling

An estimate for the expectation of f wrt. distribution $p(\cdot, \theta)$ can be obtained by sampling X wrt. a different ("proposal") distribution q:

$$\begin{aligned}
\mathcal{E}[f;\theta] &= \sum_{x \in X} f(x) p(x;\theta) \\
&= \sum_{x \in X} \underbrace{f(x) \frac{p(x;\theta)}{q(x)}}_{g(x,\theta)} q(x) \\
&\approx \frac{1}{N} \sum_{i=1}^{N} \overbrace{f(x_i) \frac{p(x_i;\theta)}{q(x_i)}} \qquad \text{where } x_i \sim q \qquad (4a) \\
&=: \hat{\mathcal{E}}[f;\theta] \qquad (4b)
\end{aligned}$$

Importance sampling

An estimate for the expectation of f wrt. distribution $p(\cdot, \theta)$ can be obtained by sampling X wrt. a different ("proposal") distribution q:

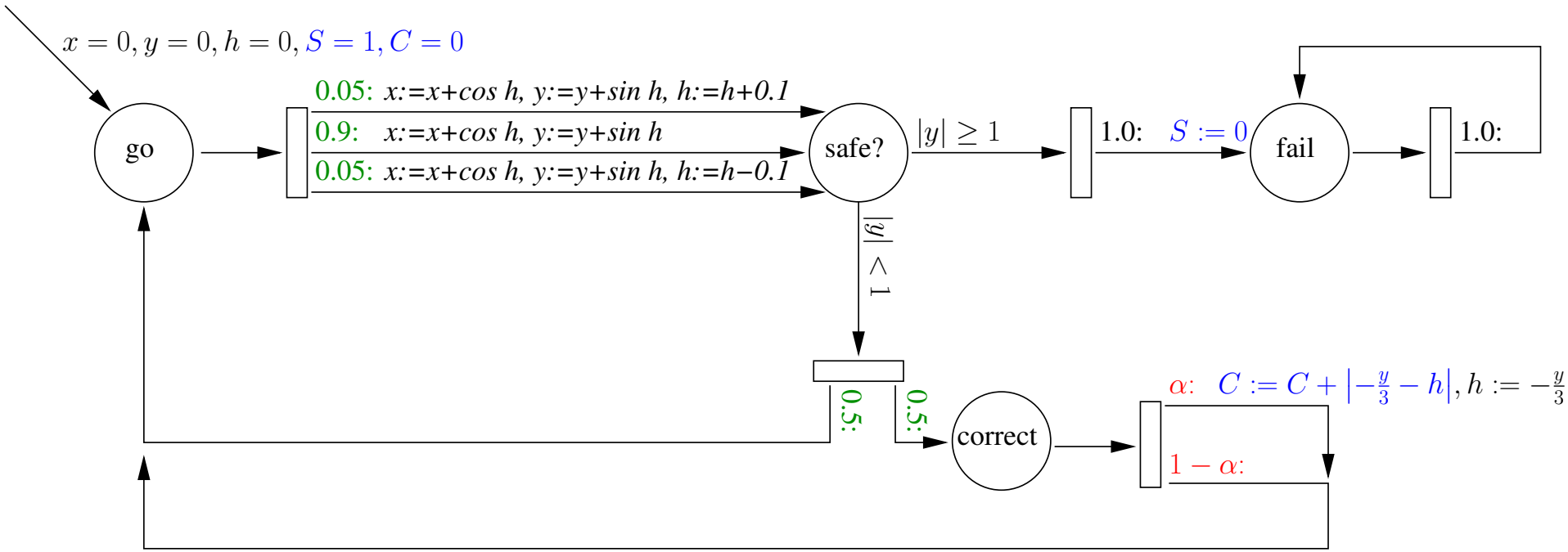
$$\begin{aligned}
\mathcal{E}[f;\theta] &= \sum_{x \in X} f(x) p(x;\theta) \\
&= \sum_{x \in X} \underbrace{f(x) \frac{p(x;\theta)}{q(x)}}_{g(x,\theta)} q(x) \\
&\approx \frac{1}{N} \sum_{i=1}^{N} \overbrace{f(x_i) \frac{p(x_i;\theta)}{q(x_i)}} \qquad \text{where } x_i \sim q \qquad (4a) \\
&=: \hat{\mathcal{E}}[f;\theta] \qquad (4b)
\end{aligned}$$

Note that samples $\{x_1, \ldots, x_N\}$ are drawn according to the substitute distribution; nevertheless, (4a–4b) permits to compute estimates $\hat{\mathcal{E}}[f;\theta]$ for arbitrary values of θ.

Symbolic Importance Sampling

Mining Symbolic Models

Determining importance weights in a PPHA



Pursue a simulation with a concrete substitute probability p replacing α.
If simulated run takes the α branch n times and the $1 - \alpha$ branch m times then, for arbitrary α,

- the probability of this run is $c \cdot p^n \cdot (1 - p)^m$ in the simulation,
- the probability of this run is $c \cdot \alpha^n \cdot (1 - \alpha)^m$ in the PPHA.

Here, c denotes the accumulated probability of all other choices along the run.

Symbolic importance sampling

$t_1, \ldots, t_l$ be the parameter-dependent probability terms in the PPHA A. Let $\#_i t_j$ denote the number of times the t_j branch was taken in run x_i when simulating A with the substitute parameterization θ^*.

Symbolic importance sampling

$t_1, \ldots, t_l$ be the parameter-dependent probability terms in the PPHA A. Let $\#_i t_j$ denote the number of times the t_j branch was taken in run x_i when simulating A with the substitute parameterization θ^*.

A symbolic representation of the parameter dependency of $\hat{\mathcal{E}}[f;\theta]$ can be obtained from importance sampling (4a–4b):

$$\hat{\mathcal{E}}[f;\theta] = \underbrace{\frac{1}{N}\sum_{i=1}^{N} f(x_i) \prod_{j=1}^{l} \left(\frac{t_j}{t_j[\theta^*/\theta]} \right)^{\#_i t_j}}_{\eta_f} \tag{5}$$

Note that $f(x_i)$, $t_j[\theta^*/\theta]$ and $\#_i t_j$ are constants s.t. the only free variables occurring in η_f are the parameters $\alpha_1, \ldots, \alpha_k$ within terms $t_1, \ldots, t_l$.

Parameter synthesis

- Term η_f in (5) is a large sum with multiple occurrences of parameters θ within different instances of sub-terms t_j.
- Let C be a constraint on the expected rewards for $\vec{f}$, i.e., C is a formula with free variable $\mathcal{E}_f$ formalizing the requirements on the expectation $\mathcal{E}[\vec{f};\theta]$.
- Let ϕ be the constraint on admissible parameterizations.

Parameter synthesis

- Term η_f in (5) is a large sum with multiple occurrences of parameters θ within different instances of sub-terms t_j.
- Let C be a constraint on the expected rewards for $\vec{f}$, i.e., C is a formula with free variable $\mathcal{E}_f$ formalizing the requirements on the expectation $\mathcal{E}[\vec{f};\theta]$.
- Let ϕ be the constraint on admissible parameterizations.

A parameter instance $\theta \models \phi$ guaranteeing C can now in principle be found — or conversely, the infeasibility of C over ϕ be established — by solving the constraint system

$$\left(\mathcal{E}_{\vec{f}} = \qquad\qquad \eta_{\vec{f}}\right) \wedge \phi \wedge C \tag{6}$$

using an appropriate constraint solver.

Parameter synthesis

- Term η_f in (5) is a large sum with multiple occurrences of parameters θ within different instances of sub-terms t_j.
- Let C be a constraint on the expected rewards for $\vec{f}$, i.e., C is a formula with free variable $\mathcal{E}_f$ formalizing the requirements on the expectation $\mathcal{E}[\vec{f};\theta]$.
- Let ϕ be the constraint on admissible parameterizations.

A parameter instance $\theta \models \phi$ guaranteeing C can now in principle be found — or conversely, the infeasibility of C over ϕ be established — by solving the constraint system

$$\left(\mathcal{E}_{\vec{f}} \in B_{\varepsilon(\delta,N)}(\eta_{\vec{f}})\right) \ \wedge \ \phi \ \wedge \ C \tag{6}$$

using an appropriate constraint solver.

Parameter synthesis

- Term η_f in (5) is a large sum with multiple occurrences of parameters θ within different instances of sub-terms t_j.
- Let C be a constraint on the expected rewards for $\vec{f}$, i.e., C is a formula with free variable $\mathcal{E}_f$ formalizing the requirements on the expectation $\mathcal{E}[\vec{f};\theta]$.
- Let ϕ be the constraint on admissible parameterizations.

A parameter instance $\theta \models \phi$ guaranteeing C can now in principle be found — or conversely, the infeasibility of C over ϕ be established — by solving the constraint system

$$\left(\mathcal{E}_{\vec{f}} \in B_{\varepsilon(\delta,N)}(\eta_{\vec{f}})\right) \ \wedge \ \phi \ \wedge \ C \tag{6}$$

using an appropriate constraint solver.

Remark: Existence of a parameter instance θ satisfying (6) is a necessary, though not sufficient condition for it satisfying the design goal with confidence. (Will deal with that issue later.)

Finding Feasible Parameter Instances

Polynomial constraint solving of very high order

The shape of the constraint formulae

- Constraint (6), i.e., $\left(\mathcal{E}_{\vec{f}} \in B_{\varepsilon(\delta,N)}(\eta_{\vec{f}})\right) \wedge \phi \wedge C$, is an arithmetic constraint containing
  1. addition, multiplication, exponentiation by integer constants,
  2. the operations found in the terms $t_1, \ldots, t_l$ defining the parameter dependency $p(\theta)$ of the Markov chain,
  3. the operations occurring in the parameter domain constraint ϕ and in the design goal C,
- it can be solved by SMT solvers addressing the corresponding subset of arithmetic, e.g. iSAT[1].

[1] iSAT is an algorithm integrating interval constraint propagation and SAT modulo theory for solving constraint systems over $\mathbb{R}, +, *, \sin, \exp, \ldots$
Implementations called HySAT II, iSAT, and iSAT-3 have been made available by the AVACS consortium from 2008 onward.

A simple instance of the constraint formulae

```
EXPR
  ...
-- X236 represents 23 sample(s) of average reward -0.434783
  X236 = -28493.9 * alpha**6 * (1-alpha)**10;
-- X235 represents 12 sample(s) of average reward -0.666667
  X235 = -21845.3 * alpha**6 * (1-alpha)**9;
-- X234 represents 35 sample(s) of average reward -0.2
  X234 = -13107.2 * alpha**9 * (1-alpha)**7;
-- X233 represents 39 sample(s) of average reward -0.0512821
  X233 = -13443.3 * alpha**7 * (1-alpha)**11;
  ...

-- Computing empirical expectation E.
  E = 0.00025 * (X1 + X2 + X3 + ... + X236 + X237 + X238 + X239);

-- Optimization target is
  (-0.01 <= E) and (E <= 0.0);

-- Parameter constraint is
  (alpha < 0.0125) or (alpha > 0.99);
```

A simple instance of the constraint formulae

```
EXPR
  ...
-- X236 represents 23 sample(s) of avera                reward -0.434783
  X236 = -28493.9 * alpha**6 * (1-alpha)
-- X235 represents 12 sample(s) of avera
  X235 = -21845.3 * alpha**6 * (1-alpha)
-- X234 represents 35 sample(s) of avera
  X234 = -13107.2 * alpha**9 * (1-alpha)
-- X233 represents 39 sample(s) of average reward
  X233 = -13443.3 * alpha**7 * (1-alpha)**11;
  ...

-- Computing empirical expectation E.
  E = 0.00025 * (X1 + X2 + X3 + ... + X

-- Optimization target is
  (-0.01 <= E) and (E <= 0.0);

-- Parameter constraint is
  (alpha < 0.0125) or (alpha > 0.99);
```

Terms over parameters can

- involve multiple different parameters,
- involve linear, polynomial, and transcendental arithmetic.

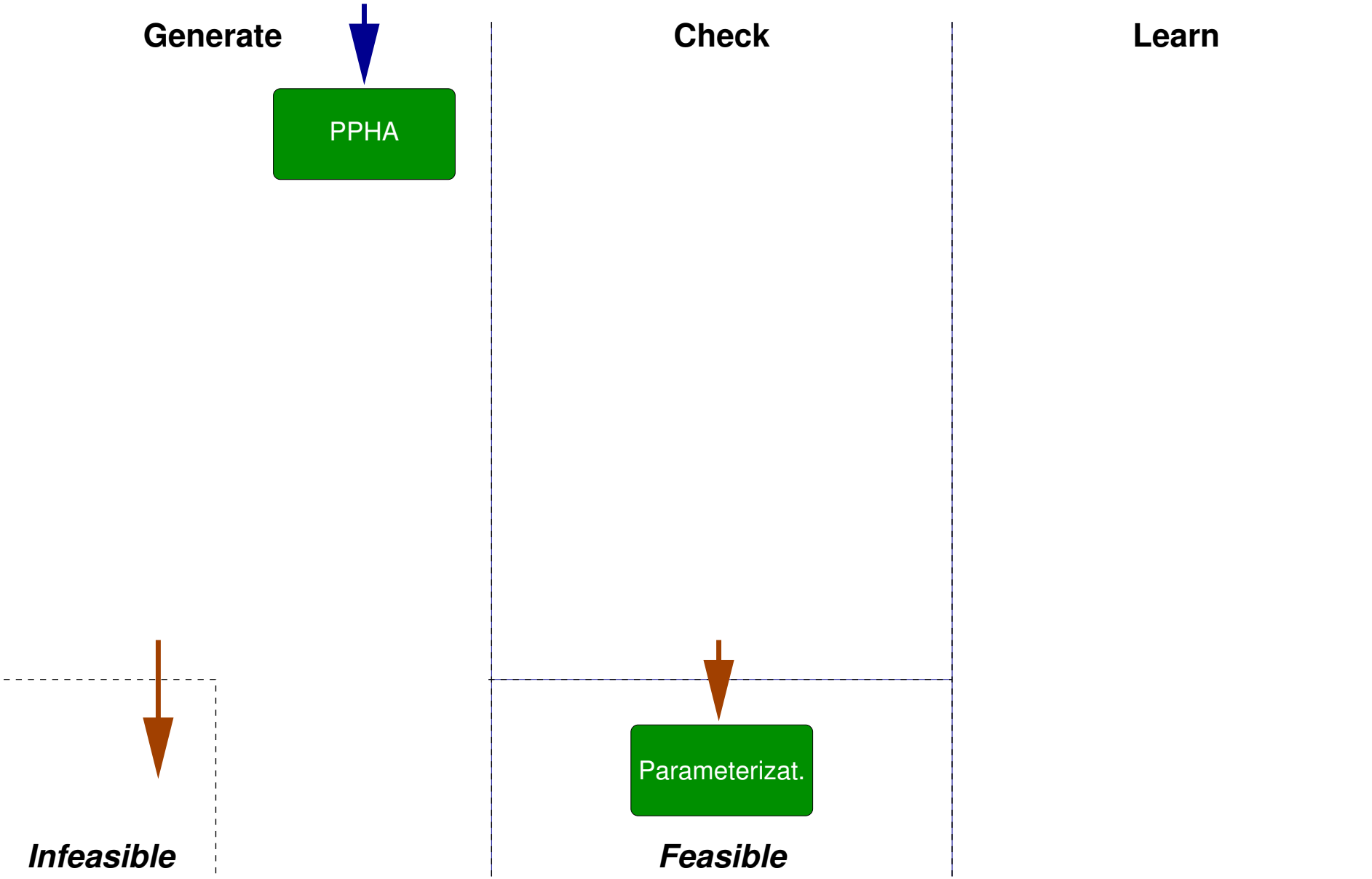
Expectations and parameters may be

- multi-dimensional,
- subject to arbitrary Boolean combinations of constraints,
- subject to non-polynomial arithmetic constraints.

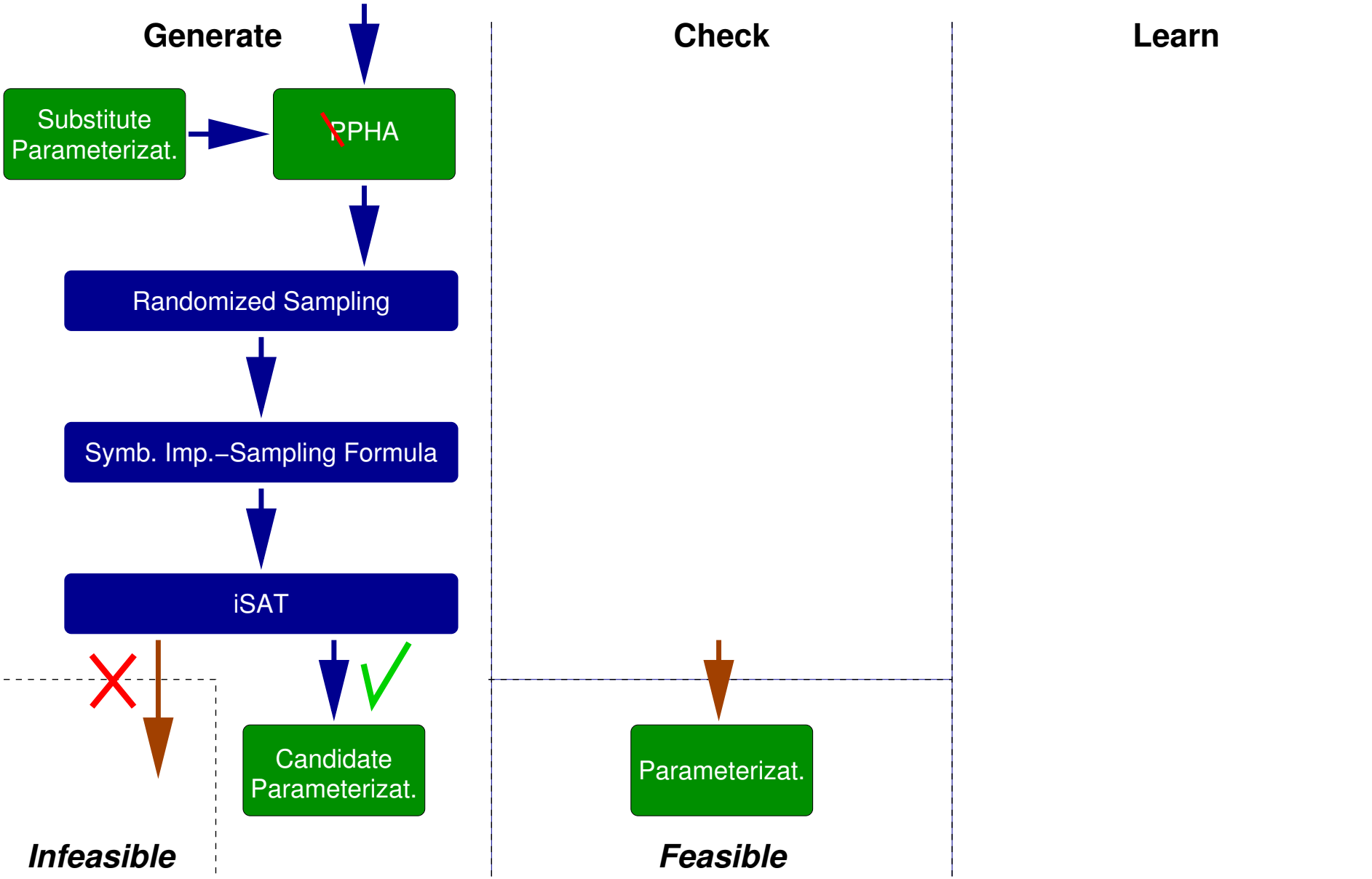
Iterative Refinement of the Symbolic Encoding

Dealing with the approximation error incurred by importance sampling

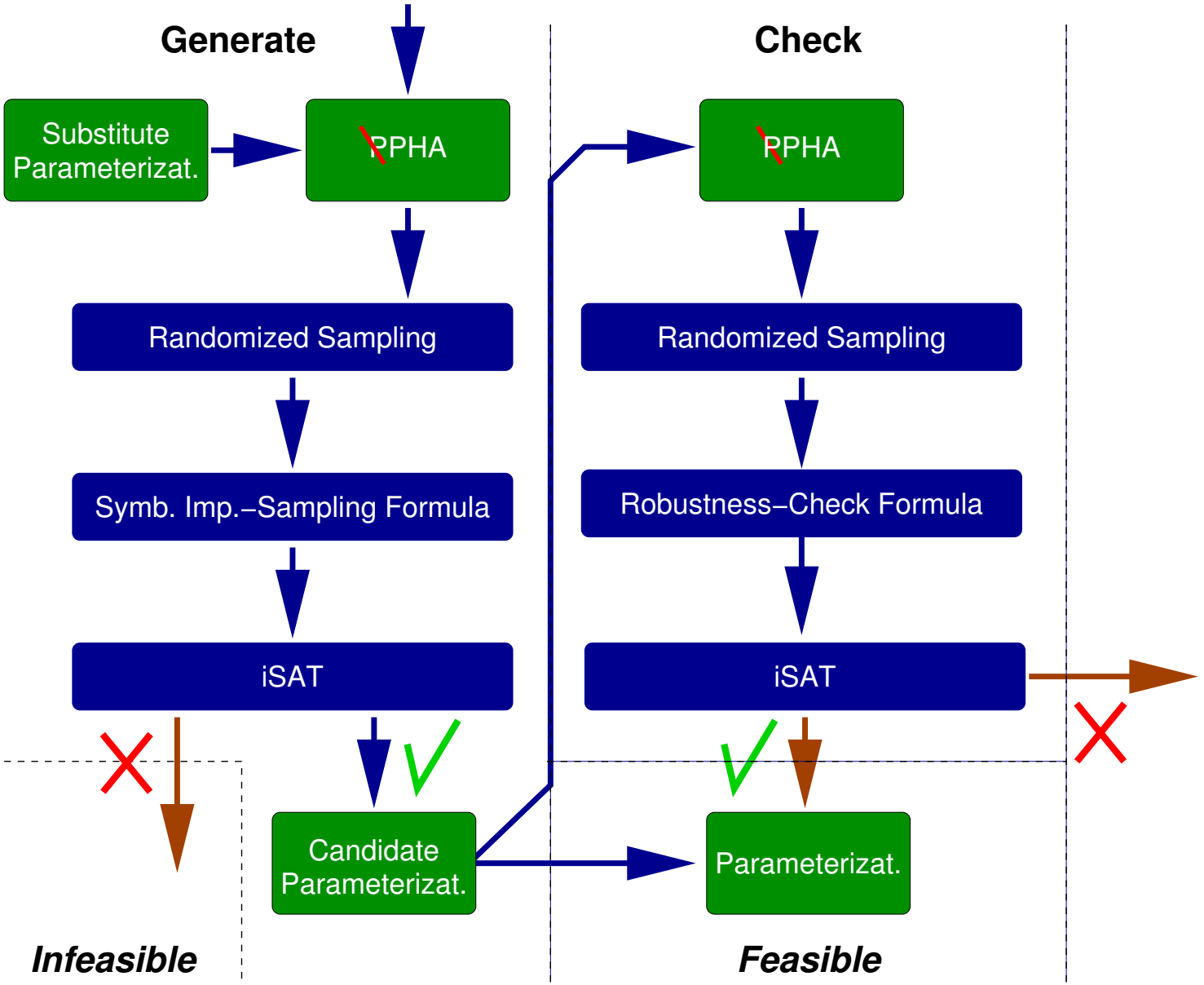
Learning from Counterexamples



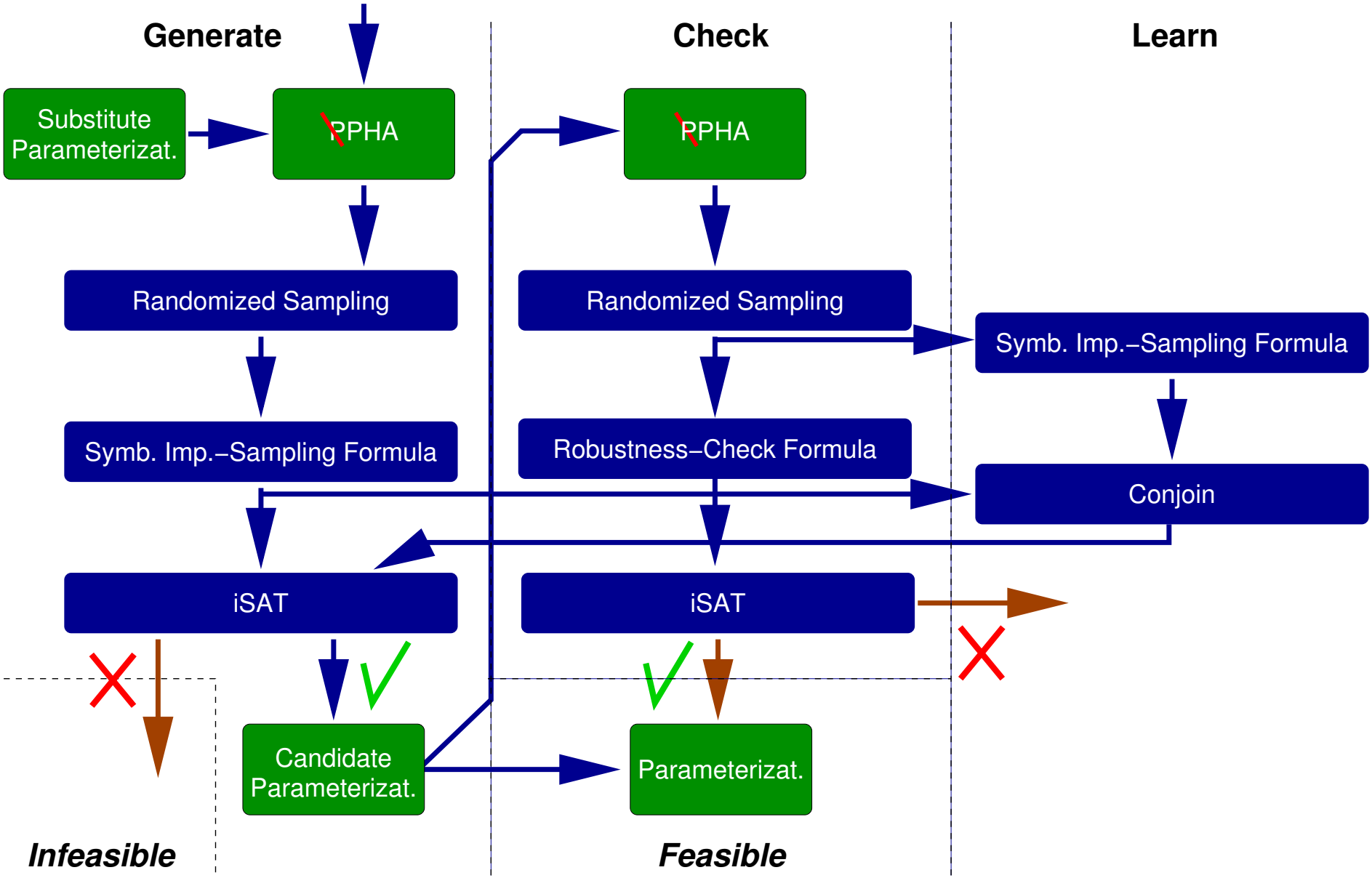
Learning from Counterexamples



Learning from Counterexamples



Learning from Counterexamples



Algorithm Properties

Let P be the user-required confidence and let the number N of samples drawn in each round be selected according to the Hoeffding bound (3).

Correctness

1. If the algorithm terminates with "Feasible" then the parameter instance provided yields expectations satisfying C with confidence $\geq P$.
2. If the algorithm terminates with "Infeasible" then for any parameter instance satisfying ϕ, the associated expectations violate C with confidence $\geq P$.

Discussion

What we did

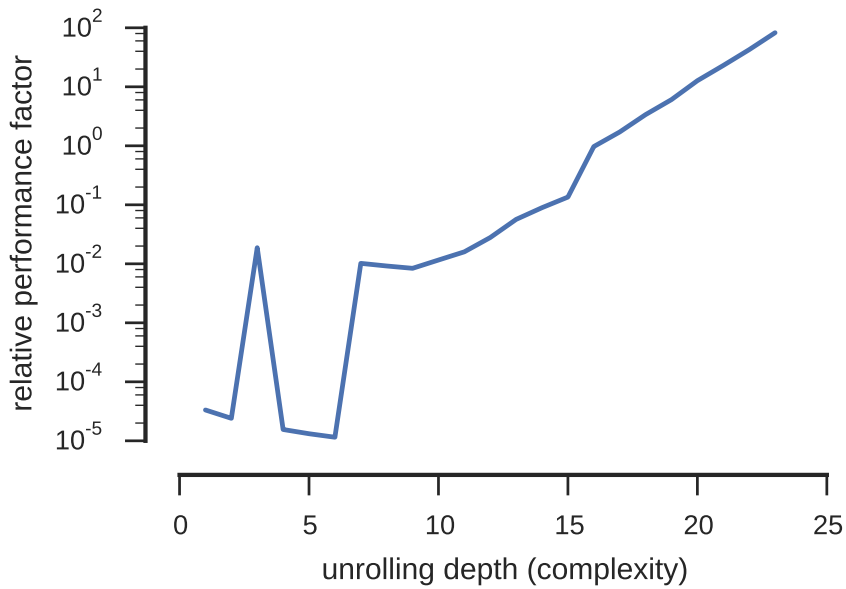
Solved a complex design-space exploration problem by
(iterative) automated learning of a tractable formal model.

- Approach is based on an alternation of *sampling*, *generalization*, *constraint generation*, *SMT solving*

What we did

Solved a complex design-space exploration problem by (iterative) automated learning of a tractable formal model.

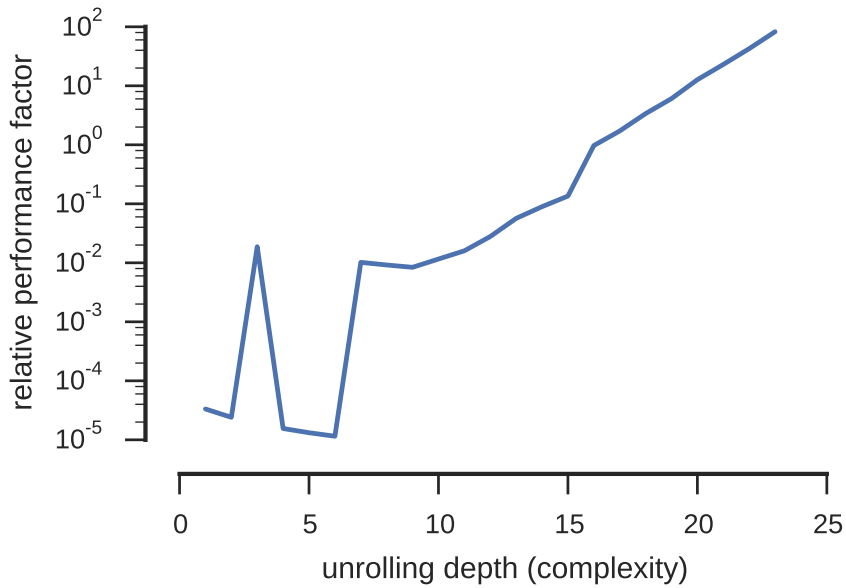
- Approach is based on an alternation of *sampling*, *generalization*, *constraint generation*, *SMT solving*
- Closed-form representation based on SMT formulae well exists, but
  - exponentially sized formulae,
  - thus infeasible.



What we did

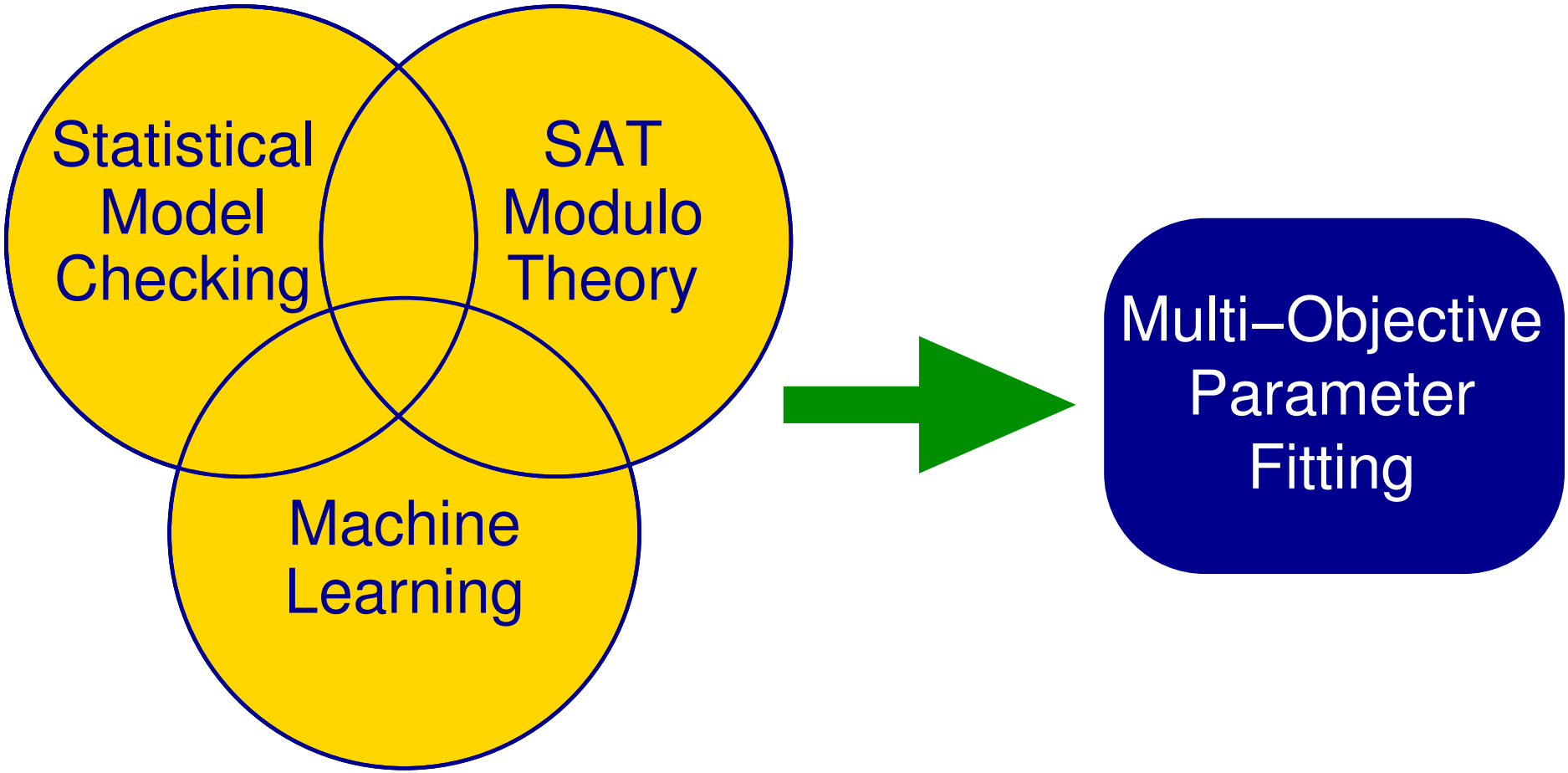
Solved a complex design-space exploration problem by (iterative) automated learning of a tractable formal model.

- Approach is based on an alternation of *sampling*, *generalization*, *constraint generation*, *SMT solving*
- Closed-form representation based on SMT formulae well exists, but
  - exponentially sized formulae,
  - thus infeasible.

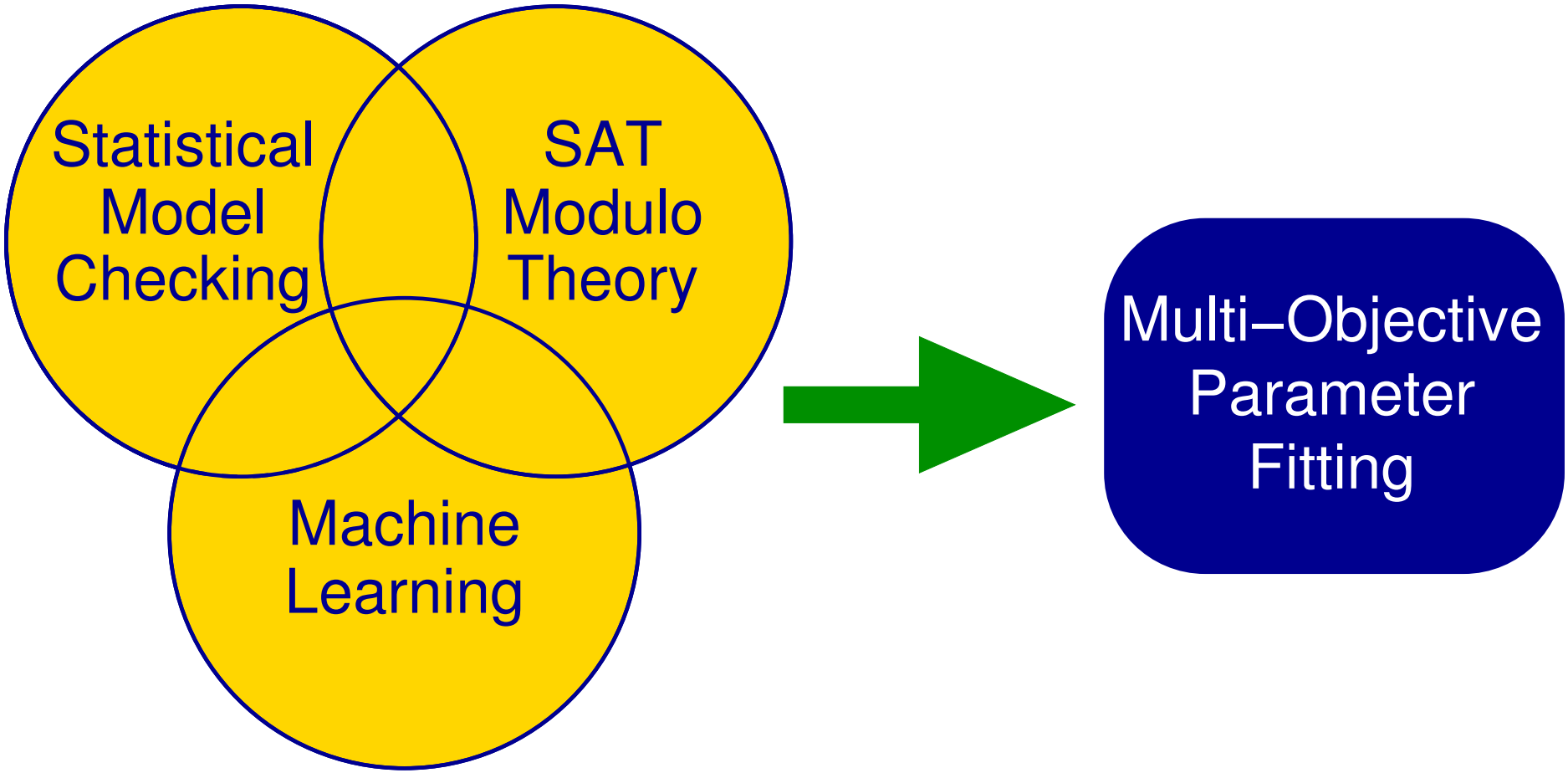


- Prototype implementation of the approach has been pursued (result of an excellent BSc thesis — thank you, Paul).

The major ingredients



The major ingredients



Many more such combinations wait to be explored!