Model Validation and Discovery for Complex Stochastic Systems

Sumit Kumar Jha

CMU-CS-10-132
July 2, 2010

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Professor Christopher J. Langmead (Chair)
Professor Robert F. Murphy
Professor Russell S. Schwartz
Professor James R. Faeder, University of Pittsburgh Medical Center
Dr. Håkan L. Younes, Google Inc.

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy.

This research was sponsored by the National Science Foundation under grant numbers CCF-0926181, CCF-0429120, and CNS-0411152; and Semiconductor Research Corporation under grant numbers 2008TJ1860 and 2005TJ1366. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government or any other entity.

Keywords: Computational Systems Biology, Computational Modeling, Model Discovery, Model Validation, Model Calibration, Temporal Logic, Behavioral Specifications, Stochastic Systems, Statistical Hypothesis Testing, Biochemical Pathways, Parameter Synthesis

Dedicated to my awesome parents and younger brother.

Abstract

In this thesis, we study two fundamental problems that arise in the modeling of stochastic systems: (i) Validation of stochastic models against behavioral specifications such as temporal logics, and (ii) Discovery of kinetic parameters of stochastic biochemical models from behavioral specifications.

We present a new Bayesian algorithm for Statistical Model Checking of stochastic systems based on a sequential version of Jeffreys' Bayes Factor test. We argue that the Bayesian approach is more suited for application domains like systems biology modeling, where distributions on nuisance parameters and priors may be known. We prove that our Bayesian Statistical Model Checking algorithm terminates for a large subclass of prior probabilities. We also characterize the Type I/II errors associated with our algorithm. We experimentally demonstrate that this algorithm is suitable for the analysis of complex biochemical models like those written in the BioNetGen language. We then argue that i.i.d. sampling based Statistical Model Checking algorithms are not an effective way to study rare behaviors of stochastic models and present another Bayesian Statistical Model Checking algorithm that can incorporate non-i.i.d. sampling strategies.

We also present algorithms for synthesis of chemical kinetic parameters of stochastic biochemical models from high level behavioral specifications. We consider the setting where a modeler knows facts that must hold on the stochastic model but is not confident about some of the kinetic parameters in her model. We suggest algorithms for discovering these kinetic parameters from facts stated in appropriate formal probabilistic specification languages. Our algorithms are based on our theoretical results characterizing the probability of a specification being true on a stochastic biochemical model. We have applied this algorithm to discover kinetic parameters for biochemical models with as many as six unknown parameters.

Acknowledgments

Thanks are due to my advisor Professor Christopher James Langmead for his constant guidance and support. The thesis would not have been possible without the support of my thesis committee members and that of the Director of the Graduate Program Professor Mor Harchol-Balter. Catherine Copetas and Deborah A. Cavlovich have provided invaluable help during my stay at Carnegie Mellon.

I am very grateful to Professor James Faeder (University of Pittsburgh Medical Center) for introducing me to rule based modeling of biochemical systems and to many challenges associated with stochastic modeling of biochemical systems. I am deeply indebted to Professor Sethu Ramesh and Dr. Swarup Mohalik of General Motors India Science Lab for many inspiring discussions on verification of embedded system models. My thanks are also due to Dr. Sriram Rajamani (Microsoft Research India) and Professor Arvinda P. Sistla (University of Illinois at Chicago) for their continued encouragement and guidance.

My sincere thanks to Professor Steve Shreve and Professor Kasper Larsen (Carnegie Mellon University) for permitting me to take their courses on Stochastic Calculus and stochastic models in finance at the Tepper School of Business. I also thank Professor Surya T. Tokdar (Duke University) for introducing me to Bayesian Data Analysis during his stay at Carnegie Mellon. Thanks are also due to Professor Bruce Krogh, Professor Xuandong Li, and Professor Goran Frehse for guiding my work on hybrid systems during my stay at Carnegie Mellon.

I am also grateful to Shengbing Jiang (GM Detroit), Thomas Fuhrman (GM Detroit), Aditya Nori (Microsoft Research India), and Prasad Naldurg (Microsoft Research India) for guiding and motivating me during my summer internships. My friends Mugizi Robert Rwebangira, Varun Gupta, Jonathan Sia, Lei Bu, Niti Garg, Ashfaque Habib and Aditya Prakash Bodicherla have made my stay in Pittsburgh pleasant over the years.

Thanks are also due to Professors A. Chakraborty, I. Sengupta, R. Mall, S. Pal , J. Mukhopadhyay, D. Sarkar, M. Sinha, C. Chakaraborti, G. Biswas, P. Chakarabati, P. Dasgupta and D. Rowchoudhary for their support during my stay at the IIT Kharagpur.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Many important phenomena in science, economics, and engineering are studied using stochastic models. Unlike deterministic models, such as ordinary differential equations, the dynamics of a stochastic model are at least partially governed by one or more random processes. For example, a model of the dynamics of eukaryotic cell division may have a deterministic component which follows an ordered sequence of state transitions (resting phase is followed by interphase, which is followed by mitosis, and then the cell returns to the resting phase), but the precise timing of these transitions may be determined by a stochastic process. The semantics of the random factor(s) will be application-dependent, but generally reflect the modeler's incomplete knowledge about certain aspects of the system (e.g., unmodeled cellular components in a cell model), the inability to control or measure certain external environmental factors (e.g., wind velocities in a vehicle dynamics model), or something more fundamental about the nature of the universe (i.e., quantum mechanics). These same random factors pose many computational challenges while working with computational models of stochastic processes. In this thesis, we consider two of these challenges — and *validation* and *model discovery*, and introduce a number of algorithms for addressing aspects of these problems in *continuous-time Markov chains* (CTMC) and *stochastic differential equations* (SDE) models. Our examples will be drawn from two application domains: computational biology and computational finance, but our algorithms and the theorems supporting them should be applicable to other domains.

In the remainder of this chapter, we briefly summarize the model validation and discovery problems, and then highlight the contributions of this thesis.

1.1 Model Validation

Model validation refers to the task of algorithmically deciding whether a given model $\mathcal{M}$ with the initial state s meets a given behavior ϕ, denoted $\mathcal{M}, s \models \phi$. For non-stochastic systems, model checking algorithms [BK08] are often used to solve the validation problem. For example, model checking has been used to verify whether a third party Windows device driver can exhibit undesirable behaviors, such as deadlock [BMMR01, BR02, NRTT09]. Unfortunately, these same algorithms cannot be applied to stochastic systems because stochastic systems are often *ergodic*, meaning that no behavior is truly impossible. Here, it becomes necessary to compute (or bound) the probability that a given behavior exists. This is most often done by performing multiple stochastic simulations which can be both costly to produce and are a poor way to investigate rare events. Thus, it is important to both minimize the expected number of samples needed to validate the model and to develop methods that are capable of investigating rare behavior. The validation algorithms presented in this thesis achieve these goals.

1.2 Model Discovery

The construction of stochastic models is usually a two step process. Certain aspects of the model like the structure of the state space model or the form of a stochastic differential equation are often obtainable from first principles. For example, our understanding of chemical reactions enables us to implicitly encode the state space using a compact set of rules [FBGH05]. Similarly, our assumptions about the behavior of a rational stock price model may lead us to the realization that geometric stochastic differential equations [BS73] are suitable models for these prices. Unfortunately, the parameters of these models (e.g., reaction rates and stock price volatilities) usually cannot be obtained from first principles. The model discovery problem is the task of finding parameters for which a given model satisfies a given specification. This problem is known to be a key challenge for modelers of stochastic systems including those in systems biology [LHFH08] and in finance [HT08]. Very often, the modeler uses her intuition to make an educated guess about the parameters and then performs extensive manual validation to discover whether her educated guess was correct. If a large number of guesses for the parameter values do not lead to a model that can be validated against the behavioral expectations from the system, the modeler is left in a dilemma: is it the case that she has not found the right parameter values yet or is the model of the stochastic system inherently flawed? The model discovery algorithms presented in this thesis automate this model discovery process for certain classes of stochastic models.

1.3 Contributions

In this thesis, we present algorithms for validation and discovery of stochastic models against high level behavioral specifications encoded in a suitable probabilistic specification language, such as probabilistic temporal logics. These specifications reflect the modeler's expectations about how the model should behave, based on domain knowledge (e.g., cell division takes approximately 2 hours) or design goals (e.g., in the context of Synthetic Biology). While we will use temporal logical specifications in our examples in this thesis, our results hold true for any specification that can be decided by observing finite-length simulations from the model.

The algorithms share a common set of inputs including: (a) an executable stochastic model $\mathcal{M}$ with parameters $\Theta = \{\theta_1, ..., \theta_n\}$, denoted $\mathcal{M}(\Theta)$, (b) a behavioral specification ϕ that encodes the expected behavior of the system, (c) the probability ρ with which $\mathcal{M}(\theta)$ should satisfy this behavioral specification, and (d) a confidence parameter T. Our model validation algorithms decide whether $\mathcal{M}(\theta), s \models P_{\geqslant\rho}(\phi)$ — that is, whether the model $\mathcal{M}(\theta)$ satisfies the property with probability at least ρ with confidence T. The model discovery algorithms find a bounded volume $\mathcal{V} \subseteq \mathbb{R}^n$ in a n-dimensional parameter space such that $\mathcal{M}(\theta), s \models P_{\geqslant\rho}(\phi)$ for any $\theta \in \mathcal{V}$. If no such volume exists (i.e., the model is infeasible), our algorithms says so and, thus, provides the modeler a proof that the model's design is flawed or incomplete.

Our validation and discovery algorithms use a new approach called *Bayesian Statistical Model Checking* that automates the process of validating stochastic models against such a probabilistic behavioral specification by implementing a sequential version of Jeffrey's

Bayes Factor test [Jef61]. We derive new theorems on the properties of our Bayesian Statistical Model Checking algorithm, including guarantees on its termination, the probability of producing an incorrect answer, and the number of sampled traces needed to achieve a given error bound. The Bayesian Statistical Model Checking also plays an important role in our model discovery algorithm. We also make novel use of survey sampling [SO96] to prove the uniform continuity and monotonicity of the probability of the property with respect to the parameters for certain classes of CTMCs.

The specific contributions of this thesis are as follows:

- The first Bayesian algorithm [JCL$^+$09] for hypothesis-testing based statistical model checking and theorems concerning that algorithm's guarantees (Chapter 4). In particular, we characterize the conditions under which the algorithm terminates, present bounds on the Type-I / Type-II errors of our algorithm, and compute the number of samples needed for the algorithm to terminate [JL10a]. The performance characteristic of the algorithm are also verified through a series of computational experiments on a variety of models.

- We then extend our Bayesian Statistical Model Checking algorithm to accommodate cost-based specifications (Chapter 4). Here, the user provides the possible loss of producing an incorrect answer and the cost of obtaining a single simulation from the stochastic model as inputs to the algorithm. The revised algorithm minimizes the overall cost of validating the model and possible loss from producing an incorrect answer. Thus, the algorithm stops when the cost of an additional simulation exceeds the expected loss due to an incorrect decision. The performance characteristic of the algorithm are also verified through a series of computational experiments.

- We then present another variation of our algorithm that uses non-independently and identically distributed (i.i.d.) sampling to investigate rare behaviors (Chapter 4) and apply it to analyze SDE models from computational finance. This algorithm preserves guarantees of termination and we provide bounds on Type I/II error and the number of samples needed by the algorithm [JL10a]. The performance characteristic of the algorithm are verified through a series of computational experiments on stochastic differential equations.

- We introduce the first algorithms for performing model discovery in a certain class of CTMCs using statistical model checking (Chapter 5). We show that when the probability that a given specification holds is uniformly continuous in the choice of parameters, it is possible to synthesize an infinite number of parameters with a finite number of experiments. We also show that this property holds for an important class of CTMCs, namely those that arise due to rule-based biochemical models, like BioNetGen [FBGH05, FBH05, FBH08]. Our algorithms are the first attempt at discovering kinetic parameters of stochastic biochemical models from high level behavioral specifications. This algorithm was first published in [JL10b].

- We show that the probability of a model satisfying a given specification is monotonic in the kinetic parameter space under mild technical conditions (Chapter 5). This permits us to construct abstractions for stochastic biochemical models and use them to discover the kinetic parameters more efficiently. We show that the monotonicity property holds for kinetic parameters in biochemical models. This algorithm was first published in [JL10b]. The performance characteristic of the algorithm are established through a series of computation experiments.

- We introduce the use of survey sampling [SO96] as a means of proving the uniform continuity of the probability of the property with respect to the parameters.

1.4 Outline of Thesis

In Chapter 2, we will summarize the kinds of stochastic models we discuss later in this thesis. We also briefly discuss the formalism in which one can write high level behavioral specifications about the model. We survey existing work on statistical model checking in Chapter 3, and also present existing work on discovery of models. In Chapter 4, we discuss our new *Bayesian Statistical Model Checking* algorithm, characterize its properties, and establish its performance characteristics through a series of computational experiments. In Chapter 5, we present our model discovery algorithms, the theorems guaranteeing correctness, and establish their performance characteristics through a series of computational experiments. We discuss conclusions from our study of validation and synthesis of complex stochastic models and survey several exciting directions for future work in Chapter 6. The Appendix contains detailed proofs of the results in this thesis.

Chapter 2

Definitions

In this chapter, we will formally define the concepts that are used throughout the thesis. In particular, we define the specific classes of stochastic models and behavioral specifications admitted by our algorithms, as well as the the model validation and discovery problems.

2.1 Stochastic Models

Stochastic models can be broadly partitioned into discrete and continuous space categories. Examples of discrete space models include Discrete Time Markov Chains (DTMCs) and Continuous Time Markov Chains (CTMCs). Discrete space models are used when the underlying phenomenon can be described in terms of a finite number of state variables, each of which have only a discrete number of possible values. In this thesis, for example, we consider biochemical systems which can be modeled as CTMCs where state variable X_i counts the number of copies of the ith molecular species. Both DTMC and CTMC models spontaneously jump from one state to another. The key difference between these models lies in how they model the passage of time, as their names suggest. Continuous state models include stochastic differential equations (SDE) and jump diffusion processes [Shr04]. In such models both the passage of time and the values of state variables are continuous. The algorithms in this thesis are limited to CTMCs and SDEs.

2.1.1 Continuous Time Markov Chains

A continuous time Markov chain (CTMC) is a stochastic model with a discrete number of states but a continuous notion of time. The system being modeled jumps from one discrete state to another while time continues to evolve continuously. We now formally define the notion of a Labeled CTMC and illustrate the definition with an example.

Definition 1 (Labeled Continuous Time Markov Chain) A labeled CTMC is a three tuple (S, L, R), where

(i) S is a *finite* set of states.

(ii) $L : S \to 2^{AP}$ is a *labeling function* that labels each state $s \in S$ with a set of atomic propositions from a finite set of atomic propositions AP.

(iii) $R : S \times S \to \mathbb{R}$ is a *rate transition matrix*. $R(s, s')$ denotes the rate of transition from state s to state s'. The notation $\mathbb{R}$ is used to denote the set of real numbers.

Example 1 *Figure 1 below illustrates a CTMC model of a simple retransmission protocol. The system goes from the initial state to the transmit state or stays at the initial state. Then, it goes from the transmission state to either a state where the transmitted packet is lost or* *the transmitted packet is successfully transmitted. From the collision state, the system will go back to the initial state. Each transition is labeled with its transition rate.*

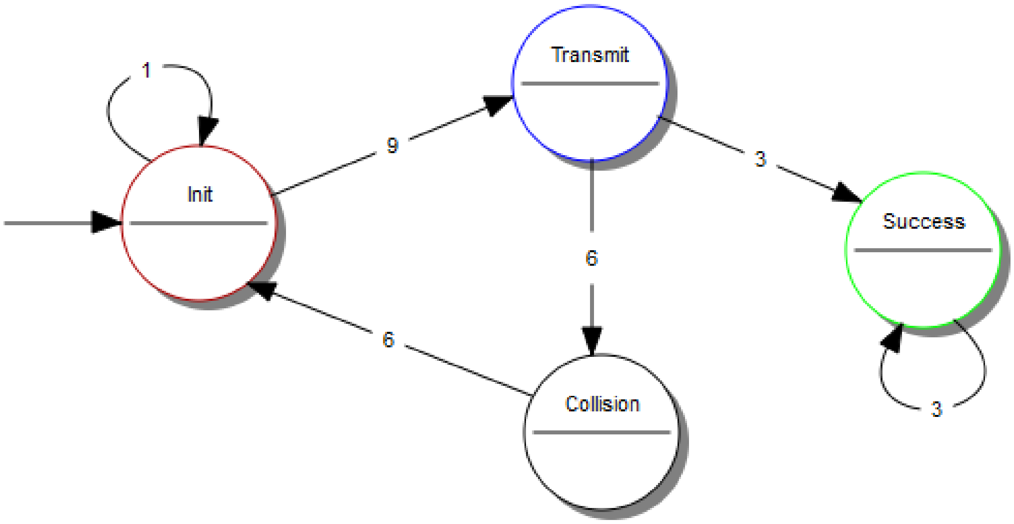


Figure 2.1: Example of a CTMC model.

The evolution of a CTMC model can be simulated by a two step process. We can first decide the time at which any of the outgoing transitions from a state should be taken. The waiting time until the first outgoing transition from a state is exponentially distributed and the rate constant of this exponential distribution is just the sum of the rates of all the outgoing transitions from this state. Having decided the time at which one of the outgoing transition is taken, we only need to decide which of the transitions should be taken. The probability of taking a transition at a given moment is propotional to the rate of the transition and the constant of proportionality is determined by the fact that probabilities of disjoing and exhaustive events (in this case, taking one of the transitions) must sum to 1. The problem with this approach is that we have to generate an exponentially distributed random number for the waiting time and a uniformly distributed random number for choosing the transition. The direct generation of exponentially distributed random samples is difficult.

In a technical report [Gil75] published at the Unites States Naval Weapons Center, Daniel T. Gillespie suggested a method to simulate the evolution of CTMC models by only using a pair of uniformly distributed random variables for each transition between states. Building upon earlier work [Doo45], Gillespie argued that the following transformation is sufficient to transform a uniformly distributed random number r into the waiting time w at a state of the CTMC:

$$w = \frac{1}{a} \log \frac{1}{r}$$

Note that a is the sum of the rates of all outgoing transitions from this state. This insight is important as it greatly reduces the amount of computational resources required for simulating CTMCs.

The semantics of a labeled CTMC model can be readily understood in terms of a *labeled path* and the probability *density* of observing such a path. We note that one cannot talk about the probability of observing a path in a CTMC as the probability of observing a particular path is zero.

Definition 2 (Labeled CTMC Path) A path σ in a labeled Continuous Time Markov Chain is a timed sequence of states $s_0 \xrightarrow{\Delta_0} s_1 \xrightarrow{\Delta_1} s_2 \cdots \xrightarrow{\Delta_{l-1}} s_l$, where $\Delta_i \geqslant 0$ is the amount of time spent in state s_i $(0 \leqslant i < l)$.

Example 2 *In Figure 1, one possible path is Init* $\xrightarrow{2.4}$ *Transmit* $\xrightarrow{1.1}$ *Success. This path corresponds to a behavior of the model where it spends time* 2.4 *units in the Init state, then moves on to the Transmit state, spends* 1.1 *time units there and then finally reaches the Success state.*

Let $E(s)$ denote the sum of the outgoing rate transitions from a state s in the CTMC (S, L, R), i.e. $E(s) = \sum_{s' \in S} R(s, s')$. The probability density $P(s, s', \Delta)$ of moving from the state s to the state s' after spending time Δ in state s is $R(s, s')\ e^{-E(s)\Delta}$. The probability density $P(\sigma)$ associated with a given path, σ, is simply the product of the probabilities of the transitions in the path. That is, $P(\sigma) = \prod_{0 \leqslant i < l} P(s_i, s_{i+1}, \Delta_i)$.

Example 3 *The probability density of the path* $\sigma \equiv$ *Init* $\xrightarrow{2.4}$ *Transmit* $\xrightarrow{1.1}$ *Success* $\circlearrowleft$ *in the CTMC represented in Figure 1 is given by*

$$\begin{aligned}
P(\sigma) &= P(\mathit{Init}, \mathit{Transmit}, 2.4)\ P(\mathit{Transmit}, \mathit{Success}, 1.1) \\
&= \left(\frac{9}{9+1} e^{-((9+1)2.4)}\right) \left(\frac{3}{3+6} e^{-((3+6)1.1)}\right) \\
&= \left(\frac{9}{10} e^{-24}\right) \left(\frac{3}{9} e^{-9.9}\right) \\
&= \frac{3}{10} e^{-33.9} \qquad\qquad \approx 5.68 \times 10^{-16}
\end{aligned}$$

2.1.2 Stochastic Differential Equations

A stochastic differential equation (SDE) [Shr04] is a differential equation in which some of the terms involve Brownian Motions. A typical SDE is of the following form:

$$dX = b(t, X_t)\ dt\ +\ v(t, X_t)\ dW_t$$

where X is a system variable, b is a Riemann integrable function, v is an Itō integrable function, and W is Brownian Motion. The Brownian Motion W is a continuous-time stochastic process satisfying the following three conditions:

(i) $W_0 = 0$

(ii) W_t is continuous (*almost surely*).

(iii) W_t has independent *normally distributed* increments:

- $W_t - W_s$ and $W_{t'} - W_{s'}$ are independent if $0 \leqslant s < t < s' < t'$.
- $W_t - W_s \sim \mathcal{N}(0, t-s)$, where $\mathcal{N}(0, t-s)$ denotes the normal distribution with mean 0 and variance $t - s$. Note that the symbol $\sim$ is used to indicate "*is distributed as*".

Consider the time between 0 and t as divided into m discrete steps $0, t_1, t_2 \ldots t_m = t$. Further, the solution to a stochastic differential equation [Shr04] is the limit of the following discrete difference equation, as m goes to infinity:

$$X_{t_{k+1}} - X_{t_k} \quad = \quad b(t_k, X_{t_k})\ (t_{k+1} - t_k)\ +\ v(t_k, X_{t_k})\ (W_{t_{k+1}} - W_{t_k})$$

Example 4 (Stochastic Models in Finance) *The price of a stock is often modeled by a geometric stochastic differential equation [HT08, BS73, Hes93, Hul06]. This is also the model for stock prices used in the famous Black-Scholes-Merton equation:*

$$dS = \mu\ S\ dt \quad + \quad \sigma\ S\ dW_t$$

Here, S represents the price of a stock, W is a Brownian Motion, μ is the constant market interest rate and σ is the constant volatility of the stock. The equation says that the rate of change in price of a stock in an infinitesimally small unit of time is the sum of a deterministic term that represents the product of the market interest rate and the current price of the stock, and a random term that depends on the inherent volatility of the market and the current price of the stock.

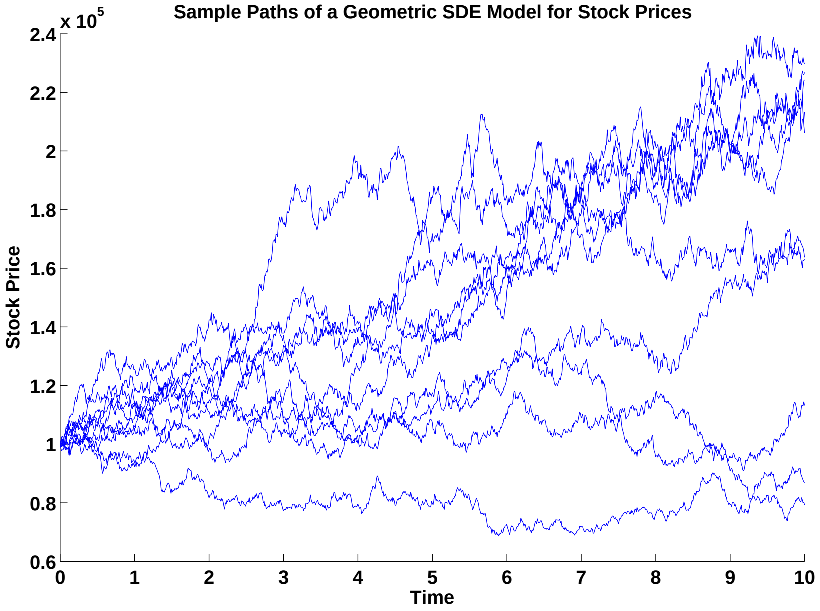


Figure 2.2: Sample Paths Obtained from an *in-silico* Stock Price Simulation

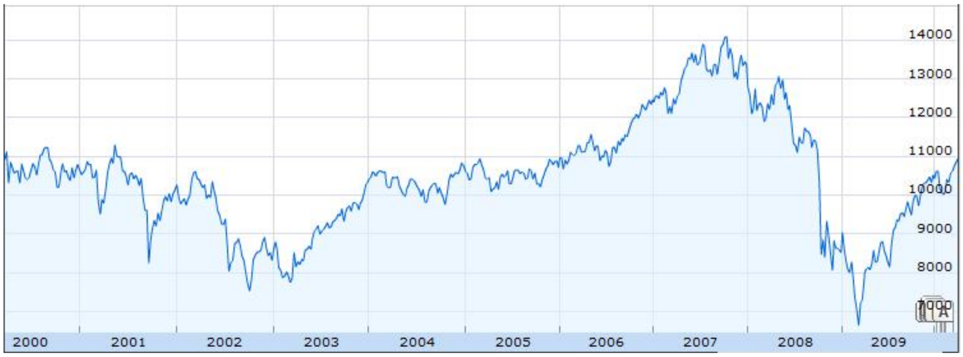


Figure 2.3: Observed Path: Dow Jones Index between 2000 and 2005

In Fig. 2.2, we observe lines representing 10 different possible future evolutions of a stock price that is worth 10 million initially. Note the qualitative similarity between the simulated plots in Fig. 2.2 and the real shape of a plot[1] *of the Dow Jones index in Fig. 2.3.*

2.2 Probabilistic Adapted Finitely Monitorable Specifications

In this section, we formally define the notion of *high-level behavioral specifications* that we can use to express the expected behavior of stochastic systems. A specification is said to be *adapted* to a stochastic process if the truth of the specification can be determined by observing a trajectories sampled from the stochastic process. Naturally, we are interested in specifications whose truth value can be decided by observing only a finite prefix of the simulation of the stochastic process. We call the logical formulae that represent such properties *adapted finitely monitorable* (AFM) specifications.

[1] Acknowledgement: This figure was produced using Google Finance

Due to the stochastic nature of our models, we generally seek to verify that a desired behavior holds with a threshold probability. Given an AFM specification ϕ, the property $Pr_{\geqslant\rho}(\phi)$ is *true* for a stochastic system if and only if the property ϕ is true for a random simulation of the stochastic system with probability at least ρ. We call such probabilistic properties *probabilistic adapted finitely monitorable* (PAFM) specifications. We survey two different kinds of PAFM specifications: Probabilistic Bounded Linear Temporal Logic (PBLTL), and Probabilistic Bounded Metric Temporal Logic (PBMTL).

Probabilistic Bounded Linear Temporal Logic

A special subclass of PAFM specifications on a stochastic model $\mathcal{M}$ can be expressed as formulas in *Probabilistic Bounded Linear Temporal Logic* (PBLTL). The logic PBLTL is similar to Continuous Stochastic Logic (CSL) [KNP04, BCHG$^+$97, BHHK03, You04] but does not permit nested probability operators. Such nestings of probability operators are not typically required to describe behaviors of interest in biological systems [LJ07a, LJC06, LJ09, JCL$^+$09, CFL$^+$08, RBFS08, Fag06, Fag05, CCD$^+$04, CF03, CFS06b]. We first define the syntax and semantics of *Bounded Linear Temporal Logic* (BLTL) [Pnu77, OL82, FS01].

For a stochastic model $\mathcal{M}$, let the set of state variables $\mathbf{V}$ be a finite set of discrete-valued variables. A Boolean predicate over $\mathbf{V}$ is a constraint of the form $x{\sim}v$, where $x \in \mathbf{V}$, $\sim \in \{\geqslant, \leqslant, =\}$, and $v \in \mathbb{R}$. A BLTL property is built on a finite set of Boolean predicates over $\mathbf{V}$ using Boolean connectives and temporal operators. The syntax of the logic is given by the following grammar:

$$\phi ::= x{\sim}v \mid (\phi_1 \vee \phi_2) \mid (\phi_1 \wedge \phi_2) \mid \neg\phi_1 \mid (\phi_1 \mathbf{U^t} \phi_\mathbf{2}),$$

where $\sim \in \{\geqslant, \leqslant, =\}$, $x \in \mathbf{V}$, $v \in \mathbb{R}$, and $t \in \mathbb{Q}_{\geqslant 0}$. We can define additional temporal operators such as $\mathbf{F^t}\psi = \mathbf{True}\,\mathbf{U^t}\,\psi$, or $\mathbf{G^t}\psi = \neg\mathbf{F^t}\neg\psi$ in terms of the bounded until $\mathbf{U^t}$. Intuitively, the formula $\mathbf{F^t}\psi$ implies that ψ holds sometime within t time units. Similarly, the formula $\mathbf{G^t}\psi$ implies that ψ holds at all moments for the next t time units into the future.

We define the semantics of BLTL with respect to the paths of $\mathcal{M}$. The fact that a path σ satisfies property ϕ is denoted by $\sigma \models \phi$. Let $\sigma = (s_0, \Delta_0), (s_1, \Delta_1), \ldots$ be an execution of the model along states $s_0, s_1, \ldots$ with durations $\Delta_0, \Delta_1, \ldots \in \mathbb{R}$. We denote the path starting at state i by σ^i (in particular, σ^0 denotes the original execution σ). The value of the state variable x in σ at the state i is denoted by $V(\sigma, i, x)$. The semantics of BLTL is defined as follows:

- $\sigma^k \models x \sim v$ if and only if $V(\sigma, k, x) \sim v$;
- $\sigma^k \models \phi_1 \vee \phi_2$ if and only if $\sigma^k \models \phi_1$ or $\sigma^k \models \phi_2$;
- $\sigma^k \models \phi_1 \wedge \phi_2$ if and only if $\sigma^k \models \phi_1$ and $\sigma^k \models \phi_2$;
- $\sigma^k \models \neg\phi_1$ if and only if $\sigma^k \models \phi_1$ does not hold (written $\sigma^k \not\models \phi_1$);
- $\sigma^k \models \phi_1 \mathbf{U}^t \phi_2$ if and only if there exists $i \in \mathbb{N}$ such that: (a) $0 \leqslant \sum_{0 \leqslant l < i} \Delta_{k+l} \leqslant t$; (b) $\sigma^{k+i} \models \phi_2$; and (c) for each $0 \leqslant j < i$, $\sigma^{k+j} \models \phi_1$.

Example 5 *Consider the following BLTL formula:*

$$\neg(G^{100}(Antibiotic > 0.05)) \vee F^{110}(Infection < 0.01)$$

It says that if the value of Antibiotic stays above 0.05 *for the first* 100 *units then the value of Infection falls below* 0.01 *within the first* 110 *units.*

It is known that finite paths of bounded duration are always sufficient for checking BLTL formula on traces [CFL$^+$08]. Given a BLTL formula, one can easily compute the duration of the prefix of the path that one may need to evaluate the formula on the entire path. We now define Probabilistic Bounded Linear Temporal Logic.

Definition 3 *A Probabilistic Bounded LTL (PBLTL) formula is a formula of the form* $P_{\geqslant \rho}(\phi)$*, where* ϕ *is a BLTL formula and* $\rho \in [0, 1]$.

We say that $\mathcal{M}$ satisfies PBLTL property $P_{\geqslant \rho}(\phi)$, denoted by $\mathcal{M} \models P_{\geqslant \rho}(\phi)$, if and only if the probability that a random path sampled from the model $\mathcal{M}$ satisfies the BLTL property ϕ is greater than or equal to ρ. This *Model Checking* problem is well-defined [YS06] since one can always assign a unique probability measure to the set of executions of $\mathcal{M}$ that satisfy a formula in BLTL.

Example 6 *Consider the following PBLTL formula:*

$$Pr_{\geqslant 0.5}\left(\neg(G^{100}(Antibiotic > 0.05)) \vee F^{110}(Infection < 0.01)\right)$$

It says that the following holds with probability at least 0.5*: if the value of Antibiotic stays above* 0.05 *for the first* 100 *units then the value of Infection falls below* 0.01 *within the first* 110 *units.*

Probabilistic Bounded Metric Temporal Logic

Another widely used logic for defining behavioral expectations from simulations of continuous (and possibly stochastic) systems is the Metric Temporal Logic [Koy90]. Monitoring algorithms [GPS09, CSV09, SS08] can automatically generate "monitors" from suitable logical specifications on simulations such that the monitor accepts a simulation of the system *if and only if* the simulation satisfies the logical specification. There exist efficient monitoring algorithms [TR05, DtS03] for Metric Temporal Logic. The logic extended with a probability operator naturally defines yet another subset of PAFM specifications.

Metric Temporal Logic (MTL) can specify both lower and upper bounds on the time bounds associated with the temporal operators. The syntax of the MTL property is given by the following grammar:

$$\phi ::= x{\sim}v \,|\, (\phi_1 \vee \phi_2) \,|\, (\phi_1 \wedge \phi_2) \,|\, \neg\phi_1 \,|\, (\phi_1 \mathbf{U}^{[\mathbf{t},\mathbf{t'}]} \phi_\mathbf{2}),$$

where $\sim \in \{\geqslant, \leqslant, =\}$, $x \in \mathbf{V}$, $v \in \mathbb{R}$, and $t \in \mathbb{Q}_{\geqslant 0}$. We can also define additional temporal operators such as $\mathbf{F}^{[\mathbf{t},\mathbf{t'}]}\psi = \mathbf{True}\,\mathbf{U}^{[\mathbf{t},\mathbf{t'}]}\,\psi$, or $\mathbf{G}^{[\mathbf{t},\mathbf{t'}]}\psi = \neg\mathbf{F}^{[\mathbf{t},\mathbf{t'}]}\neg\psi$ in terms of the bounded until $\mathbf{U}^{[\mathbf{t},\mathbf{t'}]}$. The semantics of Bounded MTL for a trace σ^k starting at the k^{th} state ($k \in \mathbb{N}$) is similar to that of BLTL except for the temporal operator $\mathbf{U}$:

- $\sigma^k \models \phi_1 \mathbf{U}^{[t,t']} \phi_2$ if and only if there exists $i \in \mathbb{N}$ such that (a) $t \leqslant \sum_{0 \leqslant l < i} \Delta_{k+l} \leqslant t'$; (b) $\sigma^{k+i} \models \phi_2$; and (c) for each $0 \leqslant j < i$, $\sigma^{k+j} \models \phi_1$.

Example 7 (t) *Consider the following Bounded Metric Temporal Logic formula:*

$$\neg(G^{[0,100]}(Antibiotic > 0.05)) \vee F^{[90,110]}(Infection < 0.01)$$

It says that if the value of Antibiotic stays above 0.05 *for the first* 100 *units then the value of Infection falls below* 0.01 *sometime between the first* 90 *units and the first* 110 *units.*

We can now define Probabilistic Bounded Metric Temporal Logic.

Definition 4 *A Probabilistic Bounded Metric Temporal Logic formula is a formula of the form* $P_{\geqslant\rho}(\phi)$*, where* ϕ *is a Bounded Metric Temporal Logic formula and* $\rho \in [0, 1]$.

We say that $\mathcal{M}$ satisfies Probabilistic Bounded Metric Temporal Logic property $P_{\geqslant\rho}(\phi)$, denoted by $\mathcal{M} \models P_{\geqslant\rho}(\phi)$, if and only if the probability that a randomly sampled execution of $\mathcal{M}$ satisfies Bounded Metric Temporal Logic property ϕ is greater than or equal to ρ.

Example 8 *Consider the following Probabilistic Bounded Metric Temporal Logic formula:*

$$Pr_{\geqslant 0.9}\left(\neg(G^{[0,100]}(Antibiotic > 0.05)) \vee F^{[90,110]}(Infection < 0.01)\right)$$

It says that the following holds with probability at least 0.5*: if the value of Antibiotic stays above* 0.05 *for the first* 100 *units then the value of Infection falls below* 0.01 *sometime between the first* 90 *units and the first* 110 *units.*

2.3 Validation of Stochastic Systems

As outlined in Chapter 1, one of the key problems addressed by this thesis is validating properties of stochastic models. Our methods combine techniques from the fields of model checking [CGP99, CE82] and Bayesian statistics, and build on the work of Younes, who invented the technique known as statistical model checking [YS06, You04, YS02, YKNP06].

Definition 5 (Model checking problem) *Given a model $\mathcal{M}$, an initial state s (or set of initial states S), and a formal specification ϕ, algorithmically decide whether $\mathcal{M}, s \models \phi$.*

Traditional model checking algorithms focus on non-stochastic models. Since this thesis concerns stochastic models, we define:

Definition 6 (Probabilistic model checking problem) *Given a stochastic model $\mathcal{M}$, an initial state s (or set of initial states S), an adapted finitely monitorable specification ϕ, and a probability bound ρ, algorithmically decide whether the $\mathcal{M}$ satisfies the specification ϕ with probability at least ρ, i.e. $\mathcal{M}, s \models P_{\geqslant \rho}(\phi)$.*

There are two basic strategies that have been employed to solve the probabilistic model checking problem. The first approach treats the problem as a probabilistic state space exploration problem [KNP04, BCHG^{+}97]. Such methods compute a *numerical estimate* of the probability that the system satisfies ϕ, and then compare the numerically computed value to ρ. The second approach, which we use in this thesis, is to use a simulation-based approach to compute a *statistical estimate* of the probability that the system satisfies ϕ. Here, a finite number of sample trajectories are drawn from the model. Each sample trajectory is evaluated to determine whether it satisfies ϕ, and the number of satisfying and non-satisfying traces is used to determine whether $\mathcal{M}, s \models P_{\geqslant \rho}(\phi)$. We will review both kinds of algorithms in the next chapter.

2.4 Discovery of Stochastic Systems

The second problem addressed in this thesis is the model discovery problem, also known as the parameter synthesis problem. Modelers often summarize domain knowledge in terms of high-level behavioral descriptions (e.g., "the system is bi-stable") and want their models to satisfy these high-level specifications with at least a certain probability. Parameter synthesis is the task of identifying the bounded volume in parameter space that gives rise to the prescribed behavior. Note that synthesis is a more challenging problem than parameter estimation, because *every* possible parameter combination must be characterized as either satisfying the property with the required probability, or not.

Definition 7 (Model discovery problem) *Given a stochastic model $\mathcal{M}$ with parameters $\theta \in \mathbb{R}^n$, a high-level behavioral specification ϕ (e.g., a formula in temporal logic), and a probability bound ρ, find the bounded volume $\mathcal{V} \subseteq \mathbb{R}^n$ in parameter space such that for any $\theta \in \mathcal{V}$, $\mathcal{M}(\theta) \models P_{\geqslant\rho}(\phi)$, and for any $\theta \notin \mathcal{V}$, $\mathcal{M}(\theta) \not\models P_{\geqslant\rho}(\phi)$*

In Chapter 5, we solve a variation of the model discovery problem for stochastic biochemical models. Instead of characterizing the exact parameter space $\mathcal{V}$ that satisfies the property ϕ with at least probability ρ, our algorithms build a parameterized approximation $\mathcal{V}_\eta$ to the actual parameter space, where $\eta > 1$. Our approximations become more precise as the parameter η gets close to 1.

Definition 8 (Approximate model discovery problem) *Given a stochastic model $\mathcal{M}$ with parameters $\theta \in \mathbb{R}^n$, a high-level behavioral specification ϕ (e.g., a formula in temporal logic), a probability bound ρ, and an approximation parameter $\eta > 1$, find the bounded* *volume* $\mathcal{V}_\eta \subseteq \mathbb{R}^n$ *in parameter space such that for any* $\theta \in \mathcal{V}$, $\mathcal{M}(\theta) \models P_{\geqslant \rho}(\phi)$*, and for any* $\theta \notin \mathcal{V}$, $\mathcal{M}(\theta) \not\models P_{\geqslant \frac{\rho}{\eta}}(\phi)$*.*

Chapter 3

Previous Work

In this chapter, we review and compare previous work in model validation and discovery with the algorithms in this thesis. Formal description and validation of computer programs has enjoyed the attention of researchers for over the last forty years [Dij76, Hoa69, Flo67, AO91, BM81]. The use of computers to build computational models of natural phenomenon and engineered systems also started around the same time [Ams70, Hay69, man63, Gil76] and is now well established [WF00, VAD98, TNO$^+$03, Sta01, SDD$^+$07, MS95, BG96, FBGH05]. However, the trend of automatically validating and even discovering computational models from behavioral specifications is relatively new [RBFS09, RBFS08, Fag06, CFS06a, LJ07b, JL10b, JCL$^+$09, CFL$^+$08]. In this chapter, we will first survey some of the techniques used for statistical validation of stochastic models against behavioral specifications encoded in a suitable logic. Then, we will discuss existing literature on the discovery of computational models from specifications.

3.1 Validation of Stochastic Systems

There are two basic approaches for solving the probabilistic model checking problem, which was defined in Section 2.3. The first approach is to treat the problem as a probabilistic state space exploration problem [KNP04, BCHG$^+$97]. Such methods compute a *numerical estimate* of the exact probability that the system satisfies ϕ using symbolic methods, and then compare the numerically computed value to ρ. Successful probabilistic model checking algorithms [BHHK03, CY95, CG04, BCHG$^+$97, KNP04] and tools [KNP04, CB06] have been proposed for various classes of models, including DTMCs, CTMCs, and Markov Decision Processes. Existing algorithms generally scale to systems with more than 10^{10} states, although in previous work we have successfully applied such algorithms to state spaces as large as 10^{23} states [LJ07a]. The scalability of these models depends on the structure of the state space and is largely unpredictable even for small models [YBO$^+$98, KNP05]. Most moderate sized biochemical models have more than 10^{100} states, and are beyond the reach of algorithms and tools based on numerical estimation.

The second approach to solving the probabilistic model checking problem is to use a statistical framework. Here, a finite number of sample trajectories are drawn from the model. Each sample trajectory is evaluated to determine whether it satisfies ϕ and the number of satisfying and non-satisfying traces is used to determine whether $\mathcal{M}, s \models P_{\geqslant\rho}(\phi)$. We refer to such strategies as *Statistical Model Checking* algorithms. The algorithms presented in Chapter 4 are statistical model checking algorithms. Statistical model checking algorithms provide approximate answers, but generally provide bounds on the probability of producing an incorrect answer. The advantage of statistical model checking algorithms over their exact counterparts is that they scale to much larger systems.

Statistical model checking algorithms can be divided into two categories: those that estimate the true value of ρ [HLMP04, Lan09], and those that treat the probabilistic model checking problem as deciding between two competing and mutually exclusive *hypotheses* [YS02, YS06, JCL$^+$09], H_0 and H_1, defined as:

$$\text{Null Hypothesis } H_0 : \mathcal{M} \models P_{\geqslant\rho}[\phi]$$

$$\text{Alternate Hypothesis } H_1 : \mathcal{M} \models P_{<\rho}[\phi]$$

Hypothesis-testing based approaches to probabilistic model checking are generally preferred over those that estimate the value of ρ because the task of deciding between two competing hypotheses is generally easier than (i.e., requires fewer samples) than obtaining an accurate estimate of ρ [You04]. The question of sample sizes is especially important in domains where the cost of generating samples is high.

Hypothesis-testing based algorithms sample a set of *independent and identically distributed* (i.i.d.) simulation traces from the model and then apply a statistical test in order to reject one of the two competing hypotheses. Given this, a *Type I error* is the rejection of the null hypothesis H_0 when it is in fact true. A *Type II error* is the rejection of the alternate hypothesis H_1 when it is true. It is generally desirable to bound the probability of making Type I and II errors. This can be done by requiring the modeler to specify an upper bound on the probability of making a wrong decision. Given this bound, it is then possible to determine the minimum number of samples required for the test to decide within the chosen error probabilities. This is called *fixed-size sampling*. Alternatively, one can iteratively draw samples from the model and terminate when enough evidence has been obtained to reject one of the hypotheses. This is called *sequential sampling*, and often leads to much smaller sample sizes because the sequential algorithm simply terminates when it has enough evidence to reject one of the hypotheses [You04].

3.1.1 Existing algorithms

Having briefly reviewed the different categories of statistical model validation algorithms, we now review the literature in more detail.

SPRT based Statistical Model Validation Younes and Simmons introduced one of the first algorithms for statistical model validation [You04, YS02, YKNP06]. They were the first to suggest the idea that statistical hypothesis testing may be used for model validation.

One of the statistical model checking algorithms suggested by Younes and Simmons is based on statistical hypothesis testing and uses Wald's *Sequential Probability Ratio Test* (SPRT) [Wal47] in particular. The SPRT decides between the null hypothesis $H'_0 : \mathcal{M} \models P_{=\rho_0}(\phi)$ against the alternate hypothesis $H'_1 : \mathcal{M} \models P_{=\rho_1}(\phi)$, where $\rho_0 \neq \rho_1$ are two probabilities. Note that these hypotheses are defined in terms of two distinct probability values, ρ_0 and ρ_1, and completely specify the distribution of the probability with which the model satisfies the formula ϕ. Such hypotheses are called *simple*. It can be shown that the SPRT is optimal for simple hypothesis testing, in the sense that it minimizes the expected number of samples among all statistical tests satisfying the same Type I and II errors [WW48] when at least one of the two hypotheses is actually true. In the case of statistical model validation, none of the two simple hypotheses may actually be true. Notice that the probabilistic model checking problem is actually a choice between two *composite* hypotheses $H_0 : \mathcal{M} \models P_{\geqslant\rho}(\phi)$ versus $H_1 : \mathcal{M} \models P_{<\rho}(\phi)$. The SPRT is known not to be optimal for composite hypotheses.

Chernoff-Hoeffding Bound based Statistical Estimation Herault et al. [HLMP04] have used the Chernoff-Hoeffding bound [Hoe63] on the sum of independent random variables to derive a fixed sample size estimator for the true value of the probability ρ with which the model $\mathcal{M}$ satisfies the specification ϕ. Given n i.i.d. Bernoulli random variables X_i such that $X_i = 1$ with some probability p and 0 otherwise, the Chernoff-Hoeffding bound gives an upper bound on the probability that the absolute difference between p and the mean of the observed samples $\frac{\sum_{i=1}^{n} X_i}{n}$ exceeds a constant $\epsilon > 0$.

$$P\left(\frac{\sum_{i=1}^{n} X_i}{n} - p \geqslant \epsilon \right) \leqslant \quad e^{-2n\epsilon^2} \tag{3.1}$$

As the number of observed samples n increases to infinity, the bound on the probability vanished to zero. Hence, it is possible to use this equation to estimate the value of the probability p with arbitrary degree of confidence. Since this approach is based on statistical estimation, it may often need a larger number of samples than the approaches based on hypothesis testing [You04]. On the other hand, it can estimate the true value with which a model satisfies a given property. One can easily sequentialize this test by re-computing Equation 3.1 after each observed sample to test the null hypothesis $H'_0 : \mathcal{M} \models P_{=\rho_0}(\phi)$ against the alternate hypothesis $H'_1 : \mathcal{M} \models P_{=\rho_1}(\phi)$, and stopping the algorithm when the probability of one of the two simple hypotheses has become sufficiently small so as to reject it. The sequential statistical test so developed is still testing simple hypotheses while the the probabilistic model checking problem is a choice between two composite hypotheses $H_0 : \mathcal{M} \models P_{\geqslant\rho}(\phi)$ versus $H_1 : \mathcal{M} \models P_{<\rho}(\phi)$. Furthermore, we already know results about the optimality of the SPRT test for simple hypotheses; hence, this sequential test based on the Chernoff-Hoeffding bound is not going to be more efficient that the SPRT test.

P-value based Statistical Model Validation Sen *et al.* [SVA04, SVA05] used the p-*value* for the null hypothesis as a frequentist statistic for hypothesis testing. The p-value is defined as the probability of obtaining observations at least as extreme as the one that was actually seen, given that the null hypothesis is true. It is important to realize that a p-value is *not* the probability that the null hypothesis is true [Goo99]. Consider a scenario where the null hypothesis is a million times as likely as the alternate hypothesis *a priori*, and the p-*value* is only 0.01. Here, the small p-value does not imply that the null hypothesis is false. Indeed, the combination of the prior knowledge about the likelihood of the hypotheses and the p-value together suggest that the null hypothesis is more likely than the alternate hypothesis.

Monte Carlo based Statistical Model Validation Grosu and Smolka have also suggested a Monte Carlo based approach for verifying formulas in LTL [GS05]. Their algorithm uses a fixed-size sampling strategy that randomly samples lassos from a Büchi automaton in an on-the-fly fashion. The algorithm terminates if it finds a counterexample. Otherwise, the algorithm provides statistical guarantees on the possible presence of a counterexample in the model. This technique has been developed for non-deterministic (and not stochastic) systems.

If the purpose of the algorithm is to detect errors in non-deterministic systems, uniform sampling algorithms are not particularly useful in detecting rare behaviors of non-deterministic systems. Also, statistical guarantees about the correctness of a non-deterministic system are not very useful as the construction of these statistical guarantees assumes a distribution on the inputs of the non-deterministic system. Simplifying assumptions on the inputs of the non-deterministic system like uniform sampling of lassos from a Büchi automaton may not hold in practice.

Bayesian Estimation based Statistical Model Validation Langmead introduced the first Bayesian technique for statistical model validation [Lan09]. This algorithm performs Bayesian estimation the mean and variance of the Bernoulli distribution modeling the probability that the formula is true. The parameters are estimated according to the following well-known formulas:

$$\hat{\rho} = \frac{k+\alpha}{\alpha+\beta+n} \qquad \hat{\nu} = \frac{(\alpha+k)(n-k+\beta)}{(\alpha+n+\beta)^2(\alpha+n+\beta+1)}$$

where $\hat{\rho}$ and $\hat{\nu}$ are the estimated mean and variance of a Bernoulli distribution after seeing n sample trajectories, of which k satisfied the formula. The prior distribution over ρ is specified in terms of the Beta distribution. α and β are the shape parameters of the Beta distribution. Our algorithm is also Bayesian, but uses hypothesis testing, and is thus expected to required fewer samples.

It is clear from this brief survey that there has been considerable research into statistical model checking but several questions still remain open:

(i) Is there a statistical model checking algorithm that works directly on the composite hypothesis testing problem posed by the probabilistic model checking problem, without reducing it to simple hypothesis tests?

(ii) Is there a Bayesian framework for performing hypothesis-based statistical model checking?

(iii) If so, can we provide frequentist guarantees for such a Bayesian algorithm?

(iv) Can we use the cost of performing a simulation and the possible loss from making an incorrect decision as the basis for a rational statistical model checking algorithm?

(v) Can we extend statistical model checking algorithms to use sampling strategies that are not i.i.d. ?

In Chapter 4, we have answered these questions affirmatively.

3.2 Discovery of Stochastic Systems

Model discovery is closely related to the problem of verification of parameterized systems, and the latter can be posed as a model checking problem. A variety of algorithms have been developed to address these problems using both symbolic [AAB00, ACH^{+}95, HHMWT00] and numerical methods [ADG05, DM07, MT00, SK03] for finite-state, continuous, and hybrid but non-stochastic systems. Such techniques differ from those presented in this thesis as they cannot be applied to stochastic models, and are restricted to safety properties.

Our algorithm for model discovery uses a combination of statistical model checking and abstraction refinement. A similar combination of techniques was first proposed in [FJK08]. That method relies on an abstraction-refinement approach [JKWC07] for model checking of linear hybrid systems which, unfortunately, cannot be easily adapted to stochastic systems.

The literature for stochastic systems is primarily focused on sensitivity analysis [GCPD05] or computing a point estimate for the parameters by fitting to observational data. Bayesian approaches to parameter estimation [JU97, Kal60, KD01, vdMDdFW00]) can be interpreted as a form of parameter synthesis because they compute a probability distribution over parameters. However, unlike our method, none of these methods admit the use of high-level behavioral specifications.

There are a number of existing algorithms that have been developed to perform parameter synthesis and related tasks for biological processes, but these are either restricted to non-stochastic models [BYWB07, CFS06a, CF03, DCS$^+$08, DCL09, DCL10, DFTdJV06, GTT03, RBFS08], or do not use high-level behavioral specifications [QBdB07]. Approximate parameter synthesis against temporal specifications has also been studied for the general class of CTMCs by using discretized parameter values and uniformization techniques [HKM08]. We also note that the notions of robustness in metric temporal logics and the use of the robustness in guiding a suitable search algorithm over the parameter space for nonlinear continuous models were first introduced by [RBFS08]. They define a metric temporal logic where a property is not just true or false, but can take an infinite continuum of values. Further, these values of the property define a metric space. The algorithm searches in the parameter space and chooses parameter values that make the model come closer to satisfying the property. The search is continued until a point in the parameter space is discovered such that the model satisfies the metric temporal logic specification. Our algorithm extends some of these concepts to the case of CTMCs.

Chapter 4

Bayesian Statistical Model Checking

In this chapter, we present our new *Bayesian Statistical Model Checking* algorithm (Sec. 4.3) and two variants (Sec. 4.4 and Sec. 4.5) based on the cost of simulating a model and the use of non-i.i.d. samples repectively. As we discussed in the previous chapters, the Probabilistic Model Checking (PMC) problem is to decide whether a model $\mathcal{M}$ satisfies an adapted finitely monitorable (*AFM*) formula ϕ with probability at least ρ, i.e. whether $\mathcal{M} \models P_{\geqslant\rho}(\phi)$, where $\rho \in [0, 1]$. Let u be the (unknown but fixed) probability of the model satisfying ϕ: thus, the PMC problem can now be re-stated as deciding between the two *composite* hypotheses about the distribution of the parameter u:

$$H'_0 : u \geqslant \rho \qquad\qquad H'_1 : u < \rho$$

These hypotheses are called *composite* because the parameter u isn't specified completely, but only in terms of a linear constraint. This may be contrasted with a *simple* hypothesis test where the distribution of the parameter u is specified completely by each of the two competing hypotheses. For example, the following is a choice between two simple hypotheses:

$$H_2 : u = \rho + \epsilon_2 \qquad\qquad H_3 : u = \rho - \epsilon_1 \qquad (0 \leqslant \rho - \epsilon_1 < \rho < \rho + \epsilon_2 \leqslant 1)$$

Our algorithm solves a slightly different version of the PMC problem. In particular, our algorithm will verify whether one of the following two relaxed hypotheses is true:

$$H_0 : u > \rho + \epsilon_2 \qquad\qquad H_1 : u < \rho - \epsilon_1 \qquad (0 \leqslant \rho - \epsilon_1 < \rho < \rho + \epsilon_2 \leqslant 1)$$

The interval $[\rho - \epsilon_1, \rho + \epsilon_2]$, where $0 \leqslant \rho - \epsilon_1 < \rho + \epsilon_2 \leqslant 1$, is called the *indifference interval* and the algorithm is permitted to accept any one of the hypotheses if the true probability actually lies in this indifference interval. Intuitively, the *null hypothesis* H_0 indicates that the model $\mathcal{M}$ satisfies the *AFM* formula ϕ with probability at least $\rho - \epsilon_1$ while the *alternate hypothesis* H_1 denotes that the model $\mathcal{M}$ does not satisfy the *AFM* formula ϕ with probability $\rho + \epsilon_2$ or more.

Recall that for any *finite* trace σ_i from a stochastic model and an adapted finitely monitorable formula ϕ, we can deterministically decide whether σ_i satisfies ϕ. Therefore, we can define a random variable X_i denoting the outcome of $\sigma_i \models \phi$. Then, X_i will be a *Bernoulli* random variable with probability mass function

$$f(x_i|u) = u^{x_i}(1-u)^{1-x_i}$$

where $x_i = 1$ *if and only if* $\sigma_i \models \phi$, otherwise $x_i = 0$. Note that the random variables X_i $(1 \leqslant i \leqslant n)$ are i.i.d.

Since the probability with which the system $\mathcal{M}$ satisfies ϕ is unknown, we can model it as a random variable U, with a density $g(u)$ called the *prior density*. The prior probability distribution is usually based on our previous experiences and beliefs about the system. A complete lack of information about the probability of the system satisfying the *AFM* formula is usually summarized by a *non-informative* or *objective* prior probability distribution [Ber85, GCSR03]. One example of a non-informative prior is Jeffreys' prior. It is a specially interesting prior as it is invariant under reparameterization of the parameter space. This suggests that Jeffreys' prior is at least independent of the specific choice of the parameterization of the parameter space.

Example 9 (Jeffreys' Prior for Bernoulli Distributions) *For a Bernoulli random variable that takes the value* 1 *with probability* ρ *and the value* 0 *with probability* $1 - \rho$*, the Jeffreys' prior for the parameter* ρ *is the Beta probability distribution with shape parameters* $1/2$ *and* $1/2$*.*

Our algorithm takes as input the prior $g(u)$, in addition to the model $\mathcal{M}$, the formula ϕ, the threshold probability ρ, and the indifference region, $[\rho - \epsilon_1, \rho + \epsilon_2]$.

4.1 Bayesian Statistics

Suppose we have a sequence of random variables $X_1, \ldots, X_n$ defined as above, and let $d = (x_1, \ldots, x_n)$ denote a sample of those variables. Recall that the *null hypothesis* H_0

indicates that the model $\mathcal{M}$ satisfies the *AFM* formula ϕ with probability at least $\rho - \epsilon_1$ while the *alternate hypothesis* H_1 denotes that the model $\mathcal{M}$ does not satisfy the *AFM* formula ϕ with probability $\rho + \epsilon_2$ or more. The *prior probability* of a hypothesis H is denoted by $P(H)$, while the *posterior probability* of a hypothesis H conditioned on the data d is denoted by $P(H|d)$. Similarly, $P(d|H)$ denotes the probability of observing the sample data $d = (x_1, \ldots, x_n)$ given that the hypothesis H is true. Now, Bayes' theorem can be used to compute the *posterior probability* in terms of *prior probability*:

$$P(H_0|d) = \frac{P(d|H_0)P(H_0)}{P(d)} \qquad\qquad P(H_1|d) = \frac{P(d|H_1)P(H_1)}{P(d)}$$

where $P(d) = P(d|H_0)P(H_0) + P(d|H_1)P(H_1)$, which in our case is always non-zero. The ratio of the posterior probability for hypotheses H_0 and H_1 given sample data d is

$$\frac{P(H_0|d)}{P(H_1|d)} = \frac{P(d|H_0)}{P(d|H_1)}\frac{P(H_0)}{P(H_1)} \tag{4.1}$$

Definition 9 *The Bayes factor $\mathcal{B}$ of sample* d *and hypotheses H_0 and H_1 is* $\mathcal{B} = \dfrac{P(d|H_0)}{P(d|H_1)}$.

The Bayes factor may be used as a measure of relative confidence in H_0 vs. H_1, as proposed by Jeffreys [Jef61, KR95]. In particular, Jeffreys suggested that a value of the Bayes factor greater than 100 provides decisive evidence in favor of H_0. Conversely, a Bayes factor less than 1/100 provides decisive evidence in favor of H_1.

In order to test H_0 vs. H_1, we compute the Bayes factor $\mathcal{B}$ of the available data and then compare it against 100 (or some other fixed threshold): we shall accept H_0 if and only if $\mathcal{B} > 100$. In Table 4.1, we present Jeffreys [Jef61] subjective interpretation to the

Bayes Factor $\mathcal{B}$	Strength of Evidence
1 - 3	Not worth more than a mention
3 - 10	Substantial
10 - 100	Strong
> 100	Decisive

Table 4.1: Jeffrey's Subjective Interpretation of Bayes Factor

values of the Bayes factor as a measure of the evidence in favor of the null hypothesis H_0:

4.2 Bayes Factor Computation

We now discuss how to numerically compute the Bayes factor. According to Definition 9, we must calculate the ratio of the probability of the observed sample $d = (x_1, \ldots, x_n)$ given the null hypothesis H_0 to that given the alternate hypothesis H_1. The probability of the observed sample given a hypothesis is obtained by integrating the joint density $h(d|u)$ of the observations with respect to the prior $g(u)$ over the interval that the hypothesis believes u to be lying in. Since, we assume that the sample is drawn from i.i.d. variables, we have that $h(d|u) = f(x_1|u)f(x_2|u) \cdots f(x_n|u)$. Therefore,

$$P(x_1, \ldots, x_n|H_0) = \int_{\rho+\epsilon_2}^{1} f(x_1|u) \ldots f(x_n|u) \cdot g(u)\, du \tag{4.2}$$

$$P(x_1, \ldots, x_n|H_1) = \int_{0}^{\rho-\epsilon_1} f(x_1|u) \ldots f(x_n|u) \cdot g(u)\, du \tag{4.3}$$

and the Bayes factor is the ratio of (4.2) and (4.3):

$$\mathcal{B} = \frac{\displaystyle\int_{\rho+\epsilon_2}^{1} f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du}{\displaystyle\int_{0}^{\rho-\epsilon_1} f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du}\,. \tag{4.4}$$

We observe that the Bayes factor depends on the data d and on the prior g, so it may be considered a measure of confidence in H_0 vs. H_1 provided by the data $x_1, \ldots, x_n$, and "weighted" by the prior g.

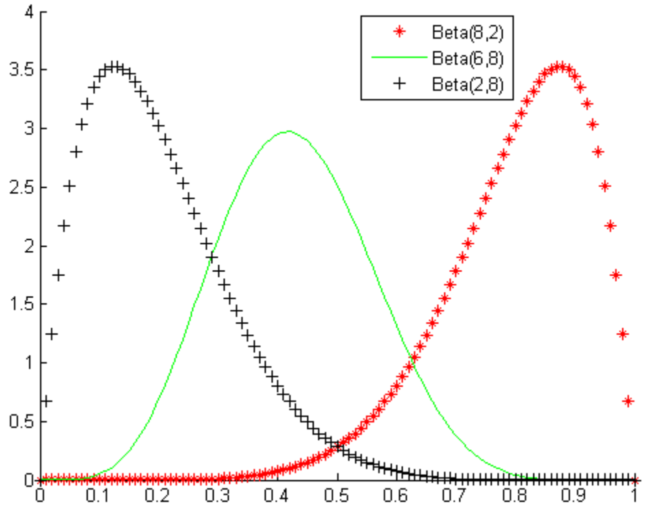


Figure 4.1: Shapes of Beta Priors

For the sake of computational convenience, it is often desirable to choose a $Beta$ distribution as the prior probability of the formula being true. The $Beta$ distribution is *conjugate* to our sampling distribution - the binomial distribution, and the choice of $Beta$ distribution as the prior yields a posterior probability distribution that can be expressed as another

$Beta$ distribution.

$$g(u) \quad = \quad \beta_{(a,b)}(u), \text{ for all values of } u \in [0,1]$$

Here, a and b are called the shape parameters of the $Beta$ distribution, and can be chosen so as to reflect different degrees of faith in the property being true (See Fig. 4.1 and Fig. 4.2).

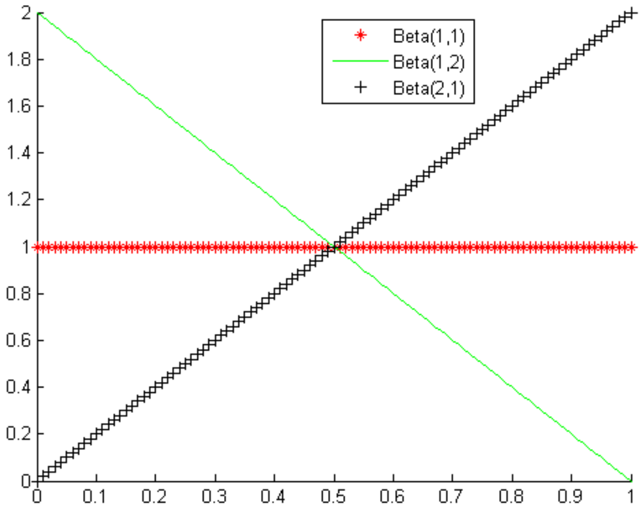


Figure 4.2: Beta Priors: Uniform Prior Beta(1,1) and Biased Priors

4.3 Algorithm 1: Bayesian Statistical Model Checking

Our Probabilistic Model Checking algorithm is essentially a *sequential* version of Jeffreys' test [Jef61]. A sequential test does not decide on the number of samples to be observed *a priori* and, instead, continues to sample more observations until one or the other hypothesis can be rejected.

Our algorithm takes five inputs:

(i) System Model $\mathcal{M}$.

(ii) A Probabilistic Adapted Finitely Monitorable (*PAFM*) Property $P_{\geqslant\rho}(\phi)$.

(iii) Indifference Region $[\rho - \epsilon_1, \rho + \epsilon_2]$ such that $(0 \leqslant \rho - \epsilon_1 < \rho < \rho + \epsilon_2 \leqslant 1)$: The algorithm is permitted to produce *any* answer if the actual probability lies in the interval $[\rho - \epsilon_1, \rho + \epsilon_2]$. The indifference regions is usually very small. Intuitively, if the true probability with which the system satisfies the AFM specification ϕ lies in this interval, we do not care about the answer to the Model Checking query. Any answer is *almost* right and hence, acceptable in this scenario.

(iv) Bayes Factor Threshold $T > 1$: The ratio of the posterior probability of one hypothesis must exceed T times the posterior probability of the other hypothesis when the algorithm stops. For example, if $T = 100$, our belief in one hypothesis must be 100 times stronger than our belief in the other hypothesis.

(v) Prior density g for the unknown parameter u: The algorithm also accepts a probability density function $g(u)$ that reflects our *prior* belief about the true probability with which the system $\mathcal{M}$ satisfies the AFM specification ϕ.

The algorithm iteratively draws i.i.d. sample traces $\sigma_1, \sigma_2, ..., \sigma_n$ from model $\mathcal{M}$ and checks whether they satisfy ϕ. As explained earlier, we can model this procedure as independent sampling from a Bernoulli distribution X of unknown parameter u - the actual probability of the model satisfying ϕ. At stage n, the algorithm has drawn samples $x_1, \ldots, x_n$. It then computes the Bayes factor $\mathcal{B}_n$ according to Equation 4.4, and it stops iff ($\mathcal{B}_n > T$ or $\mathcal{B}_n < \frac{1}{T}$). When this occurs, it will accept H_0 iff $\mathcal{B}_n > T$, and it will accept H_1 iff $\mathcal{B}_n < \frac{1}{T}$. The algorithm is illustrated in Algorithm 1.

Algorithm 1 Bayesian Statistical Model Checking

```
Require: System Model $\mathcal{M}$, Probabilistic Adapted Finitely Monitorable (PAFM) Property $P_{\geqslant\rho}(\phi)$, Indifference Region $[\rho - \epsilon_1, \rho + \epsilon_2]$, Threshold $T > 1$, Prior density $g$ for unknown parameter $u$

  $n := 0$        {total number of traces drawn}
  $x := 0$        {number of traces satisfying $\phi$}

  repeat
    $\sigma :=$ draw an i.i.d. sample trace from the system $\mathcal{M}$
    $n := n + 1$
    if $\sigma \models \phi$ then
      $x := x + 1$
    end if
    $\mathcal{B}_n :=$ BayesFactor$(n, x, \epsilon_1, \epsilon_2)$        {See Defn. (9)}
  until $(\mathcal{B}_n > T \vee \mathcal{B}_n < \frac{1}{T})$

  if $(\mathcal{B}_n > T)$ then
    return $\mathcal{M}$ satisfies the PAFM Specification $P_{\geqslant\rho}(\phi)$
  else
    return $\mathcal{M}$ does not satisfy the PAFM Specification $P_{\geqslant\rho}(\phi)$
  end if
```

4.3.1 Theorems

In this section, we prove properties of Algorithm 1. Before we do so, we recall the notions of KL divergence, affinity, δ-separation and then an important result on the concentration of Bayesian posteriors [CR08].

Definition 10 (Kullback-Leibler (KL) Divergence) *Given a parameterized family of probability distributions $\{f_\rho\}$, the Kullback-Leibler (KL) divergence $K(\rho_0, \rho)$ between the parameter distributions corresponding to the two parameters ρ and ρ_0 is $E_{\rho_0}\left[\log\left(\frac{f_\rho}{f_{\rho_0}}\right)\right]$. Note that E_{ρ_0} is the expectation computed under the probability measure f_{ρ_0}, i.e. $E_{\rho_0}[g] =$*

$\int_{-\infty}^{\infty} g\ f_{\rho_0}(x)\ dx$ for any integrable random variable g.

The KL divergence between two distributions is a measure of the difference between two probability distributions. The KL divergence is not symmetric in general and is thus not a metric.

Definition 11 (Kullback-Leibler (KL) Neighborhood) *Given a parameterized family of probability distributions $\{f_\rho\}$, the KL neighborhood $K_\epsilon(\rho_0)$ of a distribution corresponding to the parameter value ρ_0 will be given by the parameter values in the set $\{\rho : K(\rho_0, \rho) < \epsilon\}$.*

Given a parameterized family of distributions, the KL neighborhood of a distribution is the set of parameter values corresponding to distributions that are similar to this distribution under the KL divergence notion of similarity.

Definition 12 (Kullback-Leibler (KL) Support) *A parameter value ρ_0 will be said to be in the KL support of a prior Π if and only if for all $\epsilon > 0$, $\Pi(K_\epsilon(\rho_0)) > 0$.*

Given an arbitrary prior probability distribution Π, a parameter value ρ_0 is in the Kullback-Leibler (KL) support of the prior Π if and only if every *positive* KL neighborhood of ρ_0 has a non-zero prior probability.

Definition 13 (Affinity) *Let f and g be two probability distributions on a probability space R with probability measure μ. The affinity Aff(f, g) between the two densities f and g is defined as the Lebesque integral $\int_R \sqrt{fg}\ d\mu$.*

If F and G are discrete probability distributions of a random variable Y, then the above definition of affinity between F and G i.e. $\text{Aff}(F, G)$ reduces to $\sum_{y\in\mathcal{Y}} \sqrt{F(y)\ G(y)}$, where $\mathcal{Y}$ is the set of possible values of the random variable Y.

The affinity of two probability distributions is another measure of the similarity between those distributions. In particular, the affinity between two distributions is 1 if and only if they are identical. If both the distributions never together assign a non-zero probability to the same sample event, then the affinity between these two distributions is zero.

Definition 14 (Marginal Density) *For a probability measure v on ρ where $\rho \in \varrho$, $q_v^{(n)}$ is the marginal density of $X_1, \ldots, X_n$, where X_i ($1 \leqslant i \leqslant n$) are sampled i.i.d. from the distribution f_ρ.*

$$q_v^{(n)}(x_1, x_2, \ldots, x_n) = \int_{\varrho} f_\rho(x_1) \ldots f_\rho(x_n)\ v(d\rho)$$

Definition 15 (Strong δ-Separation) *Let $A \subset [0, 1]$ and $\delta > 0$. The set A and the point ρ_0 are said to be strongly δ-separated if and only if for any probability measure v on A,*

$$\mathit{Aff}\left(f_{\rho_0}, q_v^{(1)}\right) < \delta$$

Note that $q_v^{(1)}$ is the marginal density of one sample under the distribution f_ρ.

Example 10 *Let $A = [a_0, a_1]$ where $0 \leqslant a_0 < a_1 \leqslant 1$ and u be a point in $[0, 1]$ such that $u \notin A$. Let $\{f_\rho\}$ be a family of parameterized distributions. We will argue that the point u and the set A are δ-separated. We choose any probability measure v on A.*

(i) Computing the marginal density $q^{(1)}(v)$:

$$\begin{aligned} q^{(1)}(v) = & \int_{\varrho} f_\rho(x_1)\ v(d\rho) \quad \ldots \textit{ } \varrho \textit{ is the sample space} \\ = & \int_{A} f_\rho(x_1)\ v(d\rho) \quad \ldots \textit{Since, } v \textit{ is defined on } A \end{aligned}$$

(ii) Computing the affinity Aff$(f_u, q_v^{(1)})$:

$$\begin{aligned} &\textit{Aff}(f_u, q_v^{(1)}) \\ &= \int_R \sqrt{f_u \int_A f_\rho\ v(d\rho)}\ d\mu && \ldots \textit{R: sample space, } \mu\textit{: a measure on R.} \\ &= \sum_{x\in\{0,1\}} \sqrt{f_u \int_A f_\rho(.)\ v(d\rho)} && \ldots \textit{As } f \textit{ is discrete} \\ &= \sqrt{f_u(0) \int_A f_\rho(0)\ v(d\rho)} + \sqrt{f_u(1) \int_A f_\rho(1)\ v(d\rho)} && \ldots \text{Algebraic Manipulation} \\ &\leqslant \sqrt{(1-u)\ (1-a_0)} + \sqrt{u\ a_1} && \ldots\ f_\rho(0) \leqslant 1-a_0 \text{ and } f_\rho(1) \leqslant a_1 \\ &< \frac{(1-u)+(1-a_0)}{2} + \frac{u+a_1}{2} && \ldots \text{A.M.-G.M. Inequaltiy } a_0 \neq u, a_1 \neq u \\ &< 1 && \ldots u \notin A \implies f_u \neq \int_A f_\rho(x_1)\ v(d\rho). \end{aligned}$$

Hence, u and A are δ-separated for some $\delta < 1$.

Theorem 1 *If ρ_0 and $A = [a_0, a_1]$ are strongly $\delta-$separated, and $b_0 = -\log\delta$, then, for all probability v on A, for all n,*

$$\textit{Aff}\left(f(x_1|\rho_0)\ldots f(x_n|\rho_0), \int_0^1 f(x_1|u)\ldots f(x_n|u)\ v(u)du\right) < e^{-nb_0} \qquad (4.5)$$

Proof 1 *Proof by Induction:*

(i) Base Case: Aff $\Big(f(x_1|\rho_0), \int_0^1 f(x_1|u)v(u)du\Big) \leqslant \delta$*, where* $\delta < 1$*. We know this from Definition 15 and Example 10. Thus, Aff* $\Big(f(x_1|\rho_0)\ldots f(x_n|\rho_0), \int_0^1 f(x_1|u)\ldots f(x_n|u)v(u)du\Big) < e^{-b_0}$*, where* $b_0 = -\log\delta$*.*

(ii) Inductive Hypothesis: Aff $\Big(f(x_1|\rho_0)\ldots f(x_n|\rho_0), \int_0^1 f(x_1|u)\ldots f(x_n|u)v(u)du\Big) < e^{-nb_0}$*, where* $b_0 = -\log\delta$*.*

(iii) Inductive Step:

$$\begin{aligned}
&\textit{Aff}\left(f(x_1|\rho_0)\ldots f(x_{n+1}|\rho), \int_0^1 f(x_1|u)\ldots f(x_{n+1}|u)v(u)du\right)\\
&= \sum_{x\in X^{n+1}} \sqrt{f(x_1|\rho_0)\ldots f(x_{n+1}|\rho)\int_0^1 f(x_1|u)\ldots f(x_{n+1}|u)v(u)du} \quad \textit{(By Definition)}\\
&= \sum_{x\in X^{n+1}} \left(\sqrt{f(x_1|\rho_0)\ldots f(x_n|\rho_0)f(x_{n+1}|\rho)}\right.\\
&\qquad \left.\sqrt{\int_0^1 f(x_1|u)\ldots f(x_n|u)f(x_{n+1}|u)v(u)du}\right) \quad \textit{(Algebraic Manipulation)}\\
&= \sum_{x\in X^{n+1}} \left(\sqrt{f(x_1|\rho_0)\ldots f(x_n|\rho_0)f(x_{n+1}|\rho)}\right.\\
&\qquad \left.\sqrt{\int_0^1 f(x_1|u)\ldots f(x_n|u)v(u)du\int_0^1 f(x_{n+1}|u)v(u)du}\right) \quad (E[AB] = E[A]\ E[B] \textit{ if } A \perp B)\\
&= \sum_{x\in X^{n+1}} \left(\sqrt{f(x_{n+1}|\rho)\int_0^1 f(x_{n+1}|u)v(u)du}\right.\\
&\qquad \left.\sqrt{f(x_1|\rho_0)\ldots f(x_n|\rho_0)\int_0^1 f(x_1|u)\ldots f(x_n|u)v(u)du}\right) \quad \textit{(Alg. Manipulation)}\\
&= \sum_{x\in X^{n}} \sqrt{f(x_1|\rho_0)\ldots f(x_n|\rho_0)\int_0^1 f(x_1|u)\ldots f(x_n|u)v(u)du}
\end{aligned}$$

$$\sum_{x_{n+1} \in X} \sqrt{f(x_{n+1}|\rho) \int_0^1 f(x_{n+1}|u)v(u)du} \qquad \textit{(Algebraic Manipulation)}$$

$$\begin{aligned} \leqslant \quad & \mathit{Aff}\left(f(x_1|\rho_0) \ldots f(x_n|\rho_0), \int_0^1 f(x_1|u) \ldots f(x_n|u)v(u)du \right) \\ & \mathit{Aff}\left(f(x_{n+1}|\rho_0), \int_0^1 f(x_{n+1}|u)v(u)du \right) \qquad \textit{(By Definition)} \\ < \quad & e^{-nb}.e^{-b} \qquad = e^{-(n+1)b} \qquad \textit{(From base case and inductive hypothesis)} \end{aligned}$$

Hence, proved by induction.

Theorem 2 (Bayesian Consistency Theorem [CR08]) *If x_i are* i.i.d. *samples of the Bernoulli random variable $X_i (1 \leqslant i \leqslant n)$ with probability of success ρ_0 such that ρ_0 lies is in the KL support of the prior g, $A = [a_0, a_1]$ is strongly δ- separated from ρ_0 (for some $\delta > 0$), and the prior probability measure on A is finite, then the posterior probability of A decreases exponentially to 0 almost everywhere.*

$$P\left(\frac{\displaystyle\int_{a_0}^{a_1} f(x_1|u) \cdots f(x_n|u) \cdot g(u)\, du}{\displaystyle\int_0^1 f(x_1|u) \cdots f(x_n|u) \cdot g(u)\, du} \geqslant e^{-nb} \; i.o. \right) = 0$$

Here, b is a constant and the abbreviation $i.o.$ stands for infinitely often.

Proof 2 *For the details of the proof, please see [CR08]. We present a sketch of the Appendix.*

Proof of Termination

Given the preceding theorem, we can now present a general theorem that characterizes the priors under which our algorithm terminates almost surely:

Theorem 3 (Termination) *The Bayesian Statistical Model Checking algorithm (with an indifference region) terminates almost surely if the true probability with which the model $\mathcal{M}$ satisfies the formula ϕ lies in the KL support of the proper*[1] *prior probability distribution g.*

Proof 3 *There are two cases:*

(i) *Suppose the* PAFM *specification is true i.e. $\rho_0 > \rho+\epsilon_2$, then the interval $[0, \rho+\epsilon_2]$ is strongly δ-separated from ρ_0 for some non-zero constant δ (See Example 10). Then, we know that*

$$P\left(\frac{\displaystyle\int_0^{\rho+\epsilon_2} f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du}{\displaystyle\int_0^1 f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du} \geqslant e^{-nb}\ i.o.\right) = 0 \tag{4.6}$$

$$\begin{aligned}
&\Longrightarrow \frac{\displaystyle\int_0^{\rho+\epsilon_2} f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du}{\displaystyle\int_0^1 f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du} \leqslant e^{-nb}\ i.o.\ \ a.s.\\
&\Longrightarrow \frac{\displaystyle\int_0^{\rho-\epsilon_1} f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du}{\displaystyle\int_0^1 f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du} \leqslant e^{-nb}\ i.o.\ \ a.s.\ \ (\textit{Probability densities are positive})
\end{aligned} \tag{4.7}$$

[1] A probability distribution is said to be proper if the integral of the probability over any interval is a finite value.

$$\implies 1 - \frac{\displaystyle\int_0^{\rho-\epsilon_1} f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du}{\displaystyle\int_0^1 f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du} \geqslant 1 - e^{-nb} \; i.o. \;\; a.s. \;\; (\textit{Subtracting from 1})$$

$$\implies \frac{\displaystyle\int_{\rho+\epsilon_2}^1 f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du}{\displaystyle\int_0^1 f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du} \geqslant 1 - e^{-nb} \; i.o. \;\; a.s. \;\; (\textit{Algebraic Manipulation}) \tag{4.8}$$

$$\implies \frac{\displaystyle\int_{\rho+\epsilon_2}^1 f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du}{\displaystyle\int_0^{\rho-\epsilon_1} f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du} \geqslant \frac{1-e^{-nb}}{e^{-nb}} = e^{nb} - 1 \; i.o. \;\; a.s. \; (\textit{From Eqns 4.7, 4.8}) \tag{4.9}$$

The abbreviation $a.s.$ is used to indicate the phrase "almost surely". We also recall that the abbreviation $i.o.$ stands for "infinitely often".

(ii) Suppose the PAFM *specification is false i.e. $\rho_0 < \rho - \epsilon_1$, then the interval $[\rho - \epsilon_1, 1]$ is strongly δ-separated from ρ_0 for some non-zero constant δ (See Example 10). Then, we know that*

$$P\left(\frac{\displaystyle\int_{\rho-\epsilon_1}^1 f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du}{\displaystyle\int_0^1 f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du} \geqslant e^{-nb} \; i.o.\right) = 0 \tag{4.10}$$

$$\implies \frac{\displaystyle\int_{\rho-\epsilon_1}^1 f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du}{\displaystyle\int_0^1 f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du} \leqslant e^{-nb} \; i.o. \;\; a.s.$$

$$\implies \frac{\displaystyle\int_{\rho+\epsilon_2}^{1} f(x_1|u)\cdots f(x_n|u)\cdot g(u)\;du}{\displaystyle\int_{0}^{1} f(x_1|u)\cdots f(x_n|u)\cdot g(u)\;du} \leqslant e^{-nb}\; i.o.\;\; a.s. \quad (\textit{Probability densities are positive}) \tag{4.11}$$

$$\implies 1-\frac{\displaystyle\int_{\rho-\epsilon_1}^{1} f(x_1|u)\cdots f(x_n|u)\cdot g(u)\;du}{\displaystyle\int_{0}^{1} f(x_1|u)\cdots f(x_n|u)\cdot g(u)\;du} \geqslant 1-e^{-nb}\; i.o.\;\; a.s. \quad (\textit{Subtracting from 1})$$

$$\implies \frac{\displaystyle\int_{0}^{\rho-\epsilon_1} f(x_1|u)\cdots f(x_n|u)\cdot g(u)\;du}{\displaystyle\int_{0}^{1} f(x_1|u)\cdots f(x_n|u)\cdot g(u)\;du} \geqslant 1-e^{-nb}\; i.o.\;\; a.s. \quad (\textit{Algebraic Manipulation}) \tag{4.12}$$

$$\implies \frac{\displaystyle\int_{\rho+\epsilon_2}^{1} f(x_1|u)\cdots f(x_n|u)\cdot g(u)\;du}{\displaystyle\int_{0}^{\rho-\epsilon_1} f(x_1|u)\cdots f(x_n|u)\cdot g(u)\;du} \leqslant \frac{e^{-nb}}{1-e^{-nb}} = \frac{1}{e^{nb}-1}\;\; i.o.\;\; a.s. \;\; (\textit{From Eqns 4.11, 4.12}) \tag{4.13}$$

In the first case, we see that the Bayes Factor grows upwards towards infinity exponentially and in the second case, we see that the Bayes Factor shrinks down towards 0 exponentially. Hence, the algorithm always terminates.

Suppose that the specification is neither true nor false, i.e. the true probability ρ_0 lies in the indifference region $[\rho-\epsilon_1, \rho+\epsilon_2]$. We can spilt the scenario into two cases: the true probability ρ_0 lies in the region $[\rho-\epsilon_1, \rho]$ or the true probability ρ_0 lies in the region $[\rho, \rho+\epsilon_2]$.

Suppose the true probability ρ_0 lies in the region $[\rho-\epsilon_1, \rho]$. Then, the interval $[\rho+\epsilon_2, 1]$ is strongly δ-separated from ρ_0 for some non-zero constant δ.

$$P\left(\frac{\displaystyle\int_0^{\rho+\epsilon_2} f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du}{\displaystyle\int_0^1 f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du} \geqslant e^{-nb}\ i.o.\right) = 0$$

$$\Longrightarrow \frac{\displaystyle\int_0^{\rho+\epsilon_2} f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du}{\displaystyle\int_0^1 f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du} \leqslant e^{-nb}\ i.o.\ \ a.s.$$

$$\Longrightarrow \frac{\displaystyle\int_0^{\rho-\epsilon_1} f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du}{\displaystyle\int_0^1 f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du} \leqslant e^{-nb}\ i.o.\ \ a.s.\ \ \text{(Probability densities are positive)}$$

$$\Longrightarrow 1-\frac{\displaystyle\int_0^{\rho-\epsilon_1} f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du}{\displaystyle\int_0^1 f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du} \geqslant 1-e^{-nb}\ i.o.\ \ a.s.\ \ \text{(Subtracting from 1)}$$

$$\Longrightarrow \frac{\displaystyle\int_{\rho+\epsilon_2}^1 f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du}{\displaystyle\int_0^1 f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du} \geqslant 1-e^{-nb}\ i.o.\ \ a.s.\ \ \text{(Algebraic Manipulation)}$$

$$\Longrightarrow \frac{\displaystyle\int_{\rho+\epsilon_2}^1 f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du}{\displaystyle\int_0^{\rho-\epsilon_1} f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du} \geqslant \frac{1-e^{-nb}}{e^{-nb}} = e^{nb}-1\ \ i.o.\ \ a.s.$$

Thus, the Bayes Factor grows towards infinity exponentially when the true probability lies in the interval $[\rho-\epsilon_1, \rho]$.

Suppose the true probability ρ_0 lies in the region $[\rho, \rho+\epsilon_2]$. Then, the interval $[0, \rho-\epsilon_1]$ is strongly δ-separated from ρ_0 for some non-zero constant δ.

$$P\left(\frac{\displaystyle\int_{\rho-\epsilon_1}^{1} f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du}{\displaystyle\int_{0}^{1} f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du} \geqslant e^{-nb}\ i.o.\right) = 0$$

$$\implies \frac{\displaystyle\int_{\rho-\epsilon_1}^{1} f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du}{\displaystyle\int_{0}^{1} f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du} \leqslant e^{-nb}\ i.o.\ \ a.s.$$

$$\implies \frac{\displaystyle\int_{\rho+\epsilon_2}^{1} f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du}{\displaystyle\int_{0}^{1} f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du} \leqslant e^{-nb}\ i.o.\ \ a.s. \quad \text{(Probability densities are positive)}$$

$$\implies 1 - \frac{\displaystyle\int_{\rho-\epsilon_1}^{1} f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du}{\displaystyle\int_{0}^{1} f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du} \geqslant 1 - e^{-nb}\ i.o.\ \ a.s. \quad \text{(Subtracting from 1)}$$

$$\implies \frac{\displaystyle\int_{0}^{\rho-\epsilon_1} f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du}{\displaystyle\int_{0}^{1} f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du} \geqslant 1 - e^{-nb}\ i.o.\ \ a.s. \quad \text{(Algebraic Manipulation)}$$

$$\implies \frac{\displaystyle\int_{\rho+\epsilon_2}^{1} f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du}{\displaystyle\int_{0}^{\rho-\epsilon_1} f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du} \leqslant \frac{e^{-nb}}{1-e^{-nb}} = \frac{1}{e^{nb}-1}\ \ i.o.\ \ a.s.$$

Thus, the Bayes Factor shrinks towards zero exponentially when the true probability lies in the interval $[\rho, \rho+\epsilon_2]$.

Theorem 4 *When the algorithm terminates and the true probability ρ_0 of the model satisfying the specification does not lie in the indifference region $[\rho - \epsilon_1, \rho + \epsilon_2]$, the upper bound on the number of samples needed is logarithmic in the Bayes Factor threshold T.*

Proof 4 *There are two cases:*

(i) Suppose the PAFM *specification is true i.e. $\rho_0 > \rho+\epsilon_2$, then the interval $[0, \rho+\epsilon_2]$ is strongly δ-separated from ρ_0 for some non-zero constant δ (See Example 10). Thus, from Theorem 8, we know that there exist infinitely many $n_0 < \infty$ such that the following holds:*

$$P\left(\frac{\displaystyle\int_{\rho+\epsilon_2}^{1} f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du}{\displaystyle\int_{0}^{\rho-\epsilon_1} f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du} \leqslant e^{n_0 b}-1\right) = 0$$

$$\textit{i.e.}\ \frac{\displaystyle\int_{\rho+\epsilon_2}^{1} f(x_1|u)\cdots f(x_{n_0}|u)\cdot g(u)\,du}{\displaystyle\int_{0}^{\rho-\epsilon_1} f(x_1|u)\cdots f(x_{n_0}|u)\cdot g(u)\,du} \geqslant e^{n_0 b}-1\ \ a.s.$$

For the algorithm to stop, it is sufficient to pick a n_0 such that $e^{n_0 b} - 1 \geqslant T$ i.e. $n_0 \geqslant \dfrac{\log(T+1)}{b}$. Thus, the upper bound on the number of samples needed before termination is a linear function of $\log(T+1)$.

(ii) Suppose the PAFM *specification is false i.e. $\rho_0 < \rho-\epsilon_1$. Then the interval $[\rho-\epsilon_1, 1]$ is strongly δ-separated from ρ_0 for some non-zero constant δ (See Example 10). Thus, from Theorem 8, we know that there exist infinitely many $n_0 < \infty$ such that the following holds:*

$$P\left(\frac{\displaystyle\int_{\rho+\epsilon_2}^{1} f(x_1|u)\cdots f(x_{n_0}|u)\cdot g(u)\,du}{\displaystyle\int_{0}^{\rho-\epsilon_1} f(x_1|u)\cdots f(x_{n_0}|u)\cdot g(u)\,du} \geqslant \frac{1}{e^{n_0 b}-1}\right) = 0$$

$$\textit{i.e.}\ \frac{\displaystyle\int_{\rho+\epsilon_2}^{1} f(x_1|u)\cdots f(x_{n_0}|u)\cdot g(u)\,du}{\displaystyle\int_{0}^{\rho-\epsilon_1} f(x_1|u)\cdots f(x_{n_0}|u)\cdot g(u)\,du} \leqslant \frac{1}{e^{n_0 b}-1}\ \ a.s.$$

For the algorithm to stop, it is sufficient to pick a n_0 *such that* $\frac{1}{e^{n_0 b}-1} \leqslant \frac{1}{T}$ *i.e.* $n_0 \geqslant \dfrac{\log(T+1)}{b}$*. Thus, the upper bound on the number of samples needed before termination is a linear function of* $\log(T+1)$.

Bounds on Type I/II errors

We can also derive frequentist bounds on the errors. If the null hypothesis H_0 and the alternate hypothesis H_1 are mutually exclusive and γ is the ratio of the two prior probabilities i.e. $\gamma = \frac{P(H_0)}{P(H_1)}$, then the probability of the alternate hypothesis H_1 being true can be non-trivially bounded[2] if the Bayes Factor T exceeds 1. Similarly, the probability of the null hypothesis H_0 being true can be bounded non-trivially if the Bayes Factor T falls below 1. In other words, the Bayes Factor is related to the frequentist notions of Type I and Type II errors. We recall that Type I error is the probability of rejecting H_0 when the sample data has been obtained from a system for which the null hypothesis H_0 is true; similarly, Type II error is the probability of rejecting H_1 when the sample data has been obtained from a system for which the alternate hypothesis H_1 is in fact true.

[2] An upper bound larger than 1 on the value of any probability is trivial.

Theorem 5 *If the Bayes Factor threshold is T and the ratio of priors is γ, then the probability of Type-II error is at most $\dfrac{1}{T\gamma + 1}$.*

Proof 5 *We can bound the probability of the alternate hypothesis being true as follows:*

$$\begin{aligned}
&\frac{P(H_0|d)}{P(H_1|d)} && = \quad \frac{P(d|H_0)}{P(d|H_1)}\frac{P(H_0)}{P(H_1)} && \ldots \textit{From Eq. 4.1} \\
\Longrightarrow\ &\frac{P(H_0|d)}{P(H_1|d)} && \geqslant \quad T\ \ \frac{P(H_0)}{P(H_1)} && \ldots \textit{From Definition 9 and algorithm terminatio} \\
\Longrightarrow\ &\frac{P(H_0|d)}{P(H_1|d)} && \geqslant \quad T\ \ \gamma && \ldots \textit{Since, } \gamma \textit{ is ratio of priors} \\
\Longrightarrow\ &\frac{P(H_0|d)}{P(H_1|d)} + 1 && \geqslant \quad T\ \ \gamma + 1 && \ldots \textit{Adding 1 to both sides} \\
\Longrightarrow\ &\frac{P(H_0|d) + P(H_1|d)}{P(H_1|d)} && \geqslant \quad T\ \ \gamma + 1 && \ldots \textit{Simplifying} \\
\Longrightarrow\ &P(H_1|d) && \leqslant \quad \frac{P(H_0|d) + P(H_1|d)}{T\ \ \gamma + 1} && \ldots \textit{Rearranging terms} \\
\Longrightarrow\ &P(H_1|d) && \leqslant \quad \frac{1}{T\ \ \gamma + 1} && \ldots \textit{Since, } P(H_0|d) + P(H_1|d) \leqslant 1
\end{aligned}$$

We note that the probability of the alternate hypothesis being true is *inversely* proportional to two factors:

(i) The Bayes Factor threshold T : As the Bayes Factor grows larger, the bound on the probability of the alternate hypothesis being true vanishes towards 0.

(ii) The ratio of the two prior probabilities i.e. $\gamma = \frac{P(H_0)}{P(H_1)}$: As the ratio increases, the probability of the alternate hypothesis being true becomes smaller. This reflects the impact of the prior probability on our Bayesian Statistical Model Checking algorithm.

An important point to note is that the ratio of the prior probabilities γ must not be zero; otherwise, we obtain a trivial bound of 1 on the probability of the alternate hypothesis being true. In practice, this is not a problem because one always chooses a non-zero prior probability for the null hypothesis. A choice of zero prior probability for the null hypothesis would indicate that one already knows the answer to the Bayesian Statistical Model Checking query and is perfectly confident of it with probability 1.

Theorem 6 *If the Bayes Factor threshold is T and the ratio of priors is γ, then the probability of the Type-I error is at most $\dfrac{1}{\frac{T}{\gamma}+1}$.*

Proof 6 *We can bound the probability of the null hypothesis being true as follows:*

$$
\begin{array}{llll}
& \dfrac{P(H_0|d)}{P(H_1|d)} & = & \dfrac{P(d|H_0)}{P(d|H_1)}\dfrac{P(H_0)}{P(H_1)} & \ldots \textit{From Eq. 4.1} \\
\Longrightarrow & \dfrac{P(H_0|d)}{P(H_1|d)} & \leqslant & \dfrac{1}{T}\ \dfrac{P(H_0)}{P(H_1)} & \ldots \textit{From Definition 9 and algorithm termination} \\
\Longrightarrow & \dfrac{P(H_0|d)}{P(H_1|d)} & \leqslant & \dfrac{1}{T}\ \gamma & \ldots \textit{Since, } \gamma \textit{ is ratio of priors} \\
\Longrightarrow & \dfrac{P(H_1|d)}{P(H_0|d)} & \geqslant & \dfrac{T}{\gamma} & \ldots \textit{Inverting both sides} \\
\Longrightarrow & \dfrac{P(H_1|d)}{P(H_0|d)}+1 & \geqslant & \dfrac{T}{\gamma}+1 & \ldots \textit{Adding 1 to both sides} \\
\Longrightarrow & \dfrac{P(H_1|d)+P(H_0|d)}{P(H_0|d)} & \geqslant & \dfrac{T}{\gamma}+1 & \ldots \textit{Simplifying} \\
\Longrightarrow & P(H_0|d) & \leqslant & \dfrac{P(H_1|d)+P(H_0|d)}{\frac{T}{\gamma}+1} & \ldots \textit{Rearranging terms} \\
\Longrightarrow & P(H_0|d) & \leqslant & \dfrac{1}{\frac{T}{\gamma}+1} & \ldots \textit{Since, } P(H_1|d)+P(H_0|d) \leqslant 1
\end{array}
$$

We note that the probability of the null hypothesis being true is *directly* proportional to two factors:

(i) The Bayes Factor threshold T : As the Bayes Factor grows smaller towards 0, the bound on the probability of the null hypothesis being true vanishes towards 0.

(ii) The ratio of the two prior probabilities i.e. $\gamma = \frac{P(H_0)}{P(H_1)}$: As the ratio increases, the probability of the null hypothesis being true becomes larger. This reflects the impact of the prior probability on our Bayesian Statistical Model Checking algorithm.

An important point to note is that the ratio of the prior probabilities γ must not be infinite. Otherwise, the prior probability of the alternate hypothesis being true is 0. Thus, one already knows the answer to the Bayesian Statistical Model Checking query and is perfectly confident of it with probability 1.

In Table 4.3.1, we also relate Bayes Factor with Type-I and Type-II errors. We note that the Type-I and II errors are also influenced by the ratio of the priors. Note that our analysis does not bound both the Type-I and Type-II errors at the same time. One of the two bounds turns out to be trivial as the bound on the probability is larger than 1 and we know that all probabilities are trivially bounded by 1.

Ratio of Priors γ	Bayes Factor T	Type-I error	Type-II error
1	100	$\frac{1}{101}$	No Bound
1	0.01	No Bound	$\frac{1}{101}$
100	100	$\frac{1}{10001}$	No Bound
100	0.01	No Bound	$\frac{1}{2}$
0.01	100	$\frac{1}{2}$	No Bound
0.01	0.01	No Bound	$\frac{1}{10001}$

Table 4.2: Illustrating the Frequentist Properties of Bayes Factor Test

4.3.2 Empirical Performance of Algorithm 1

The performance of the Bayesian Statistical Model Checking algorithm depends on five factors:

(i) the actual probability with which the system satisfies the PAFM formula (ρ_0),

(ii) the probability threshold (ρ) in the PAFM formula,

(iii) the Bayes threshold (T),

(iv) the width of the indifference region, and

(v) the Prior Probability density (g).

It is independent of any other details of the precise model being studied. In this section, we will study the performance of our algorithm in terms of these five factors.

Influence of the difference between ρ and ρ_0

In Figure 4.3, we study the number of samples needed by our algorithm for a system $\mathcal{M}$ which satisfies a specification ϕ with probability $\rho_0 = 0.25$. We chose a symmetric indifference region of size 0.001 and a Bayes Factor of 100. We repeated each experiment 100 times and have reported the average number of samples needed.

We see that the number of samples needed by our algorithm increases as the threshold probability in our formula ρ gets closer to the actual probability of the formula being true ρ_0 . The same trend is also observed in Figures. 4.4 and 4.5, where we study systems that satisfy the specification ϕ with probability $\rho_0 = 0.5$ and $\rho_0 = 0.75$ respectively. Thus, the number of required samples is inversely proportional to $|\rho - \rho_0|$.

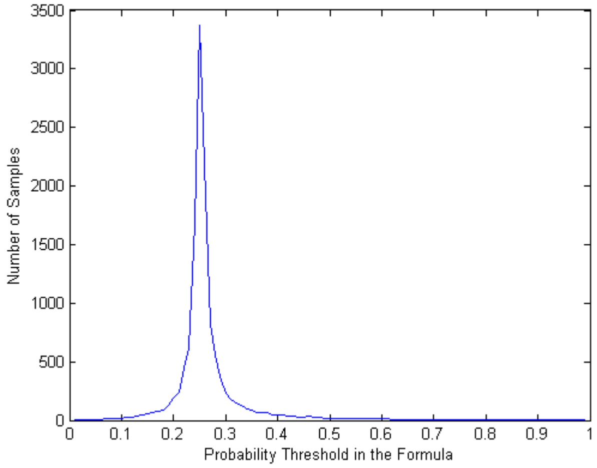


Figure 4.3: Threshold Probability (ρ) vs. Number of Samples. Actual Probability (ρ_0) is 0.25.

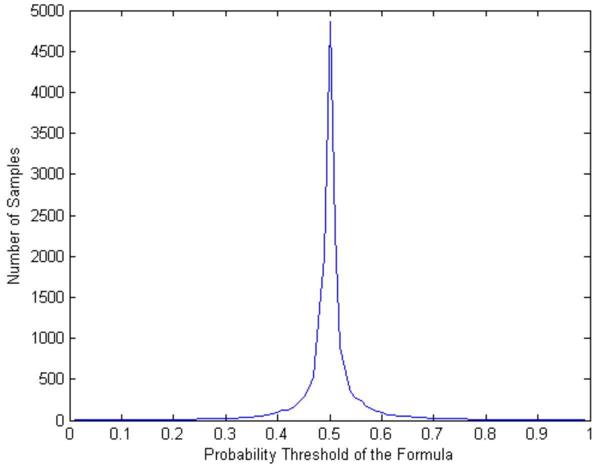


Figure 4.4: Threshold Probability (ρ) vs. Number of Samples. Actual Probability (ρ_0) is 0.5.

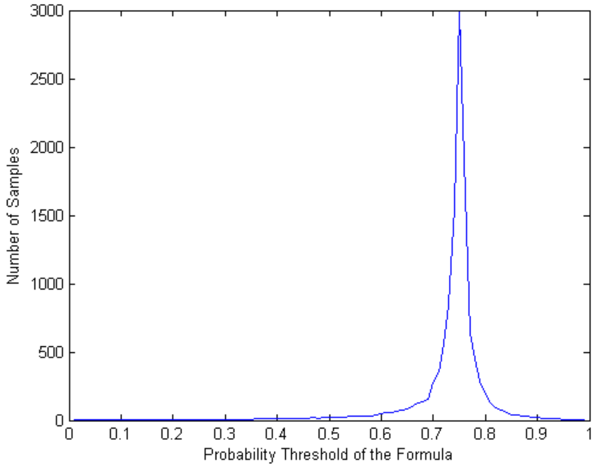


Figure 4.5: Threshold Probability (ρ) vs. Number of Samples. Actual Probability (ρ_0) is 0.75.

Influence of Bayes Factor Threshold (T) on the Performance of the algorithm

In Figure 4.6, we study the number of samples needed by our algorithm for a system $\mathcal{M}$ that satisfies a specification ϕ with probability $\rho_0 = 0.75$. We test a property that holds on the system with probability $\rho = 0.5$. We chose a symmetric indifference region of size 0.001 and a Bayes Factor of 100. We repeated each experiment 10000 times and have reported the average number of samples needed.

We find that the logarithm of the Bayes factor threshold and the number of samples needed for our algorithm to terminate have a linear relationship, as predicted by Theorem 4. We see that the linear relationship also holds in Figures. 4.7 and 4.8, where we study systems where the property holds on the system with probability $\rho_0 = 0.25$ and $\rho_0 = 0.9$ respectively. In Fig. 4.9, we study the case when the true probability of the formula being

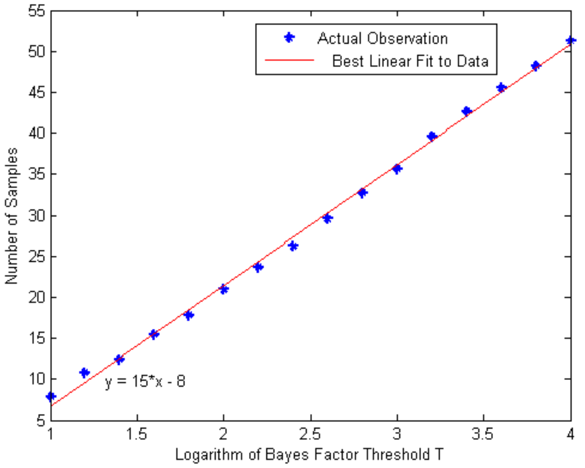


Figure 4.6: Logarithm of the Bayes Factor Threshold (T) vs. Number of Samples. Probability Threshold (ρ) is 0.5.

true lies in the indifference region, and find that the number of samples is still linearly related to the logarithm of the Bayes Factor threshold.

Influence of Indifference Region on the Performance of the algorithm

In Figure 4.10, we study the number of samples needed by our algorithm for a system $\mathcal{M}$ that satisfies a specification ϕ with probability $\rho_0 = 0.75$. We test a property that holds on the system with probability $\rho = 0.8$. We chose a Bayes factor threshold of 100. We repeated each experiment 1000 times and have reported the average number of samples needed.

As expected, we find that a smaller indifference regions leads to an increase in the number of the samples needed by our algorithm. However, as the indifference region con-

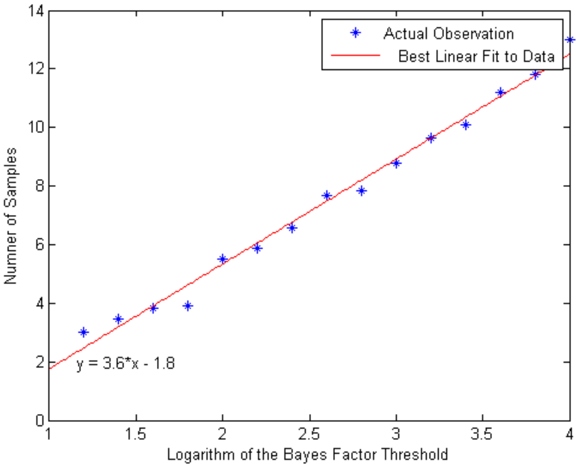


Figure 4.7: Logarithm of the Bayes Factor Threshold (T) vs. Number of Samples. Probability of Satisfying the Formula (ρ_0) is 0.25.

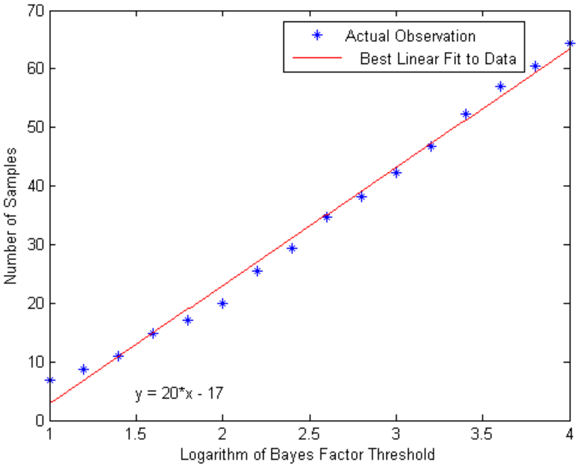


Figure 4.8: Logarithm of the Bayes Factor Threshold (T) vs. Number of Samples. Probability of Satisfying the Formula (ρ_0) is 0.9.

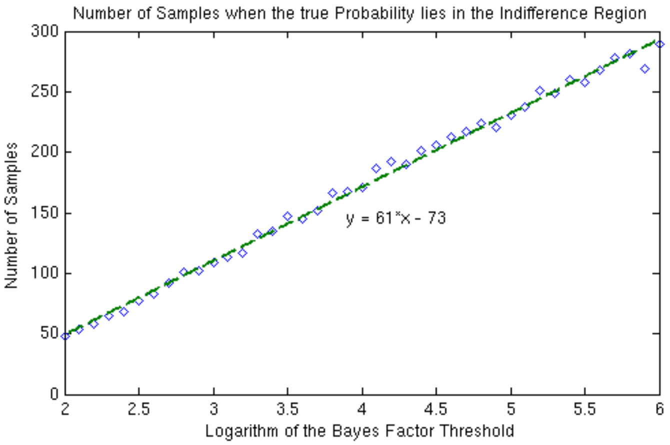


Figure 4.9: Logarithm of the Bayes Factor Threshold (T) vs. Number of Samples. Probability of Satisfying the Formula (ρ_0) is 0.8. The indifference region is the interval $[0.65, 0.85]$ and the probability in the specification is 0.75.

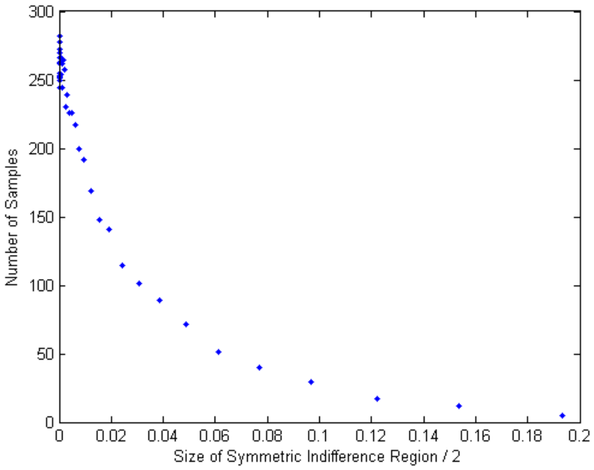


Figure 4.10: Indifference Region vs. Number of Samples

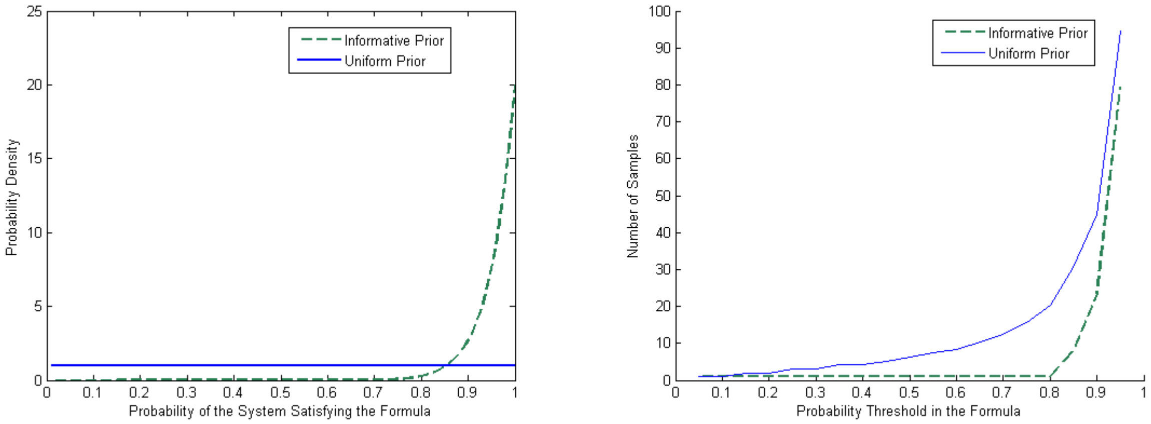
tinues to shrink even further, the number of samples needed by the algorithm approaches a constant asymptotically. This is also expected as the algorithm would terminate even with a 0 indifference region as long as the true probability ρ_0 of the system satisfying the formula is not identical to the threshold probability ρ in the PAFM specification.

Performance with varying priors

We investigated the effect of priors on the performance of the Bayesian Model Checking algorithm. We used two different priors - non-informative prior and an informative prior. The priors and the number of samples needed by the Bayesian algorithm for these priors is plotted in Fig. 4.11(a) and Fig. 4.11(b). The priors used are Beta distributions with different shape parameters: (i) $\alpha = 1/2, \beta = 1/2$: non-informative prior, (ii) $\alpha = 20, \beta = 1$: informative prior with a peak around 0.99.

We study the number of samples needed by our algorithm for a property which says that $\mathcal{M}$ which satisfies a specification ϕ with probability $\rho_0 = 0.99$. We chose a symmetric indifference region of size 0.01 and a Bayes Factor of 100. We repeated each experiment 100 times and have reported the average number of samples needed. For each choice of prior probability, we varied the threshold probability (ρ) in the specification and recorded the number of samples needed by our algorithm.

Fig. 4.11(b) shows that the number of samples needed by the Bayesian algorithm becomes smaller when the prior probability distribution is informative and supports the true hypothesis. A completely non-informative prior does not perform as well as an informative prior. However, the algorithm still performs quite well, particularly when the actual



(a) Shape of the Priors used in our Experiments.

(b) Number of Samples with Different Classes of Priors.

Figure 4.11: Different Classes of Priors

probability of the system is away from the threshold probability in the formula.

Performance of the Algorithm when the Threshold Probability is the Actual Probability of the Formula being true

An interesting scenario for Bayesian Statistical Model Checking arises when the threshold probability ρ in the PAFM specification is identical to the actual probability ρ_0 of the formula being true.

We ran our algorithm multiple number of times on a system which satisfies a property with probability 0.75. The threshold probability we chose was also 0.75, and we used a symmetric indifference region of 0.01 on each side. In Figure 4.12, we plot the Bayes Factor when the algorithm terminates for various runs of the Bayesian Statistical Model Checking algorithm with these parameters.

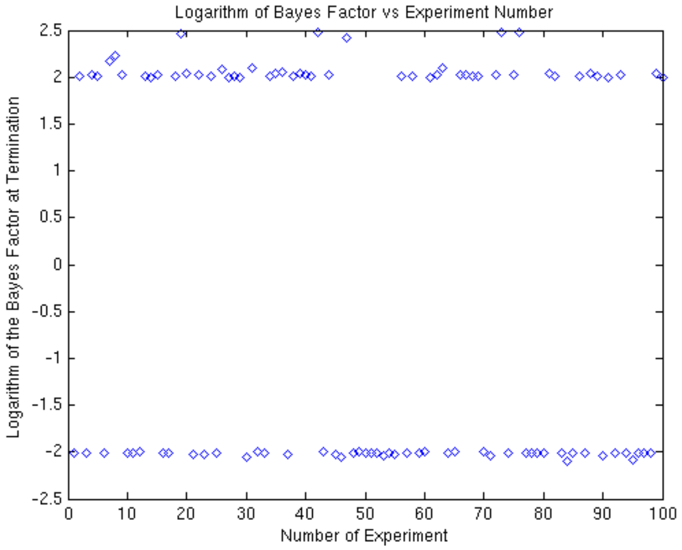


Figure 4.12: Bayes Factor at Termination.

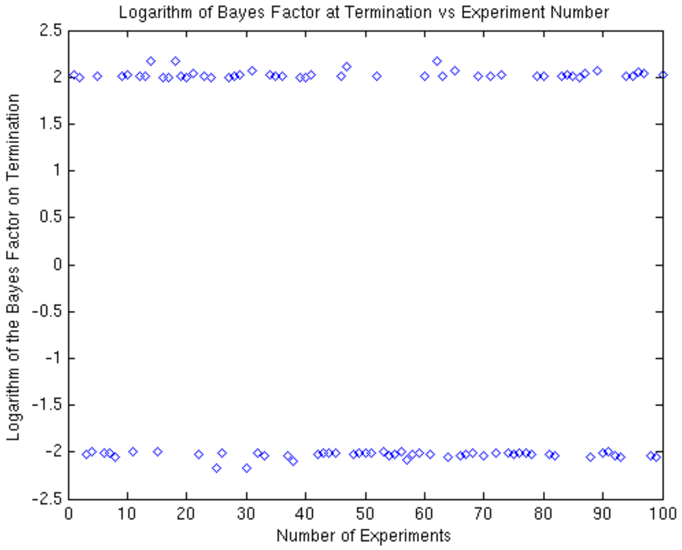


Figure 4.13: Bayes Factor at Termination.

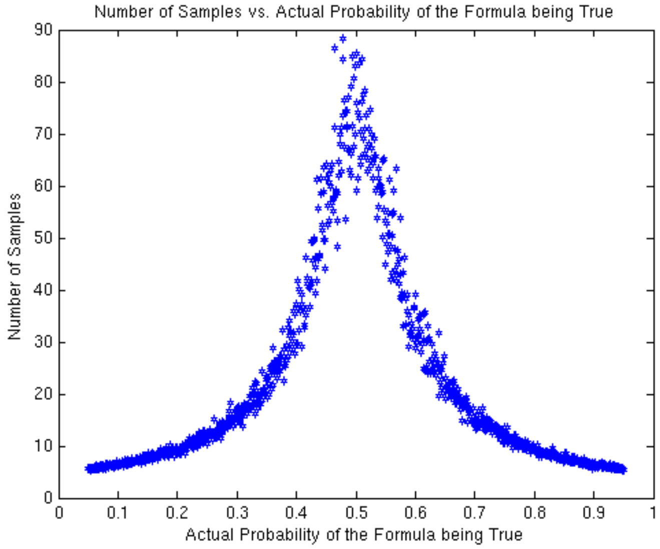


Figure 4.14: Number of Samples vs. Probability of the Formula being True

We also ran our algorithm multiple number of times on another model which satisfies a property with probability 0.5. The threshold probability we chose was also 0.5, and we used a symmetric indifference region of 0.01 on each side. In Figure 4.13, we plot the Bayes Factor when the algorithm terminates for various runs of the Bayesian Statistical Model Checking algorithm with these parameters.

Influence of the difference between the True Probability and the Threshold Probability on the Performance of the Algorithm

We also experimentally studied the difference between the true probability of the property being true on the model and the threshold probability of the specification.

In Fig. 4.14, we study the number of samples needed for the Bayesian Statistical Model

Checking algorithm to terminate when the threshold probability is 0.5 and we vary the actual probability of the formula being true from 0.05 to 0.95. The shape of the curve in the figure indicates that the number of samples is inversely related to the exponential of the distance between the threshold probability and the actual porbability of the specification being true on the model.

Case Study: T Cell Receptor Pathway

The previous experiments were on synthetic systems. In this section we apply Algorithm 1 to a model of the T cell receptor pathway. T lymphocytes, also known as T cells, play a central role in the immune system by detecting foreign substances, known as antigens, and coordinating the immune response. T cells detect the presence of antigen through surface receptors, called T cell receptors (TCRs), which bind to specific polypeptide fragments that are displayed on the surface of neighboring cells by a protein called the major histocompatibility complex (MHC). Variable regions of the immunoglobulin chains that comprise the TCR give rise to a broad range of TCR binding specificities. Individual T cells (or clonal populations derived from the same precursor) express a unique form of TCR. Processes of positive and negative selection during maturation of T cells in the thymus select T cells possessing TCRs with a weak but nonzero affinity for binding MHC molecules carrying peptides derived from host proteins. High-affinity binding between TCR and peptide-MHC (pMHC) complexes induces a cascade of biochemical events that leads to activation of the T cell and initiation of an immune response. To be effective in detecting antigens while avoiding autoimmunity, T cells must generate strong responses to the presence of minute quantities of antigen—as low as a few peptide fragments per

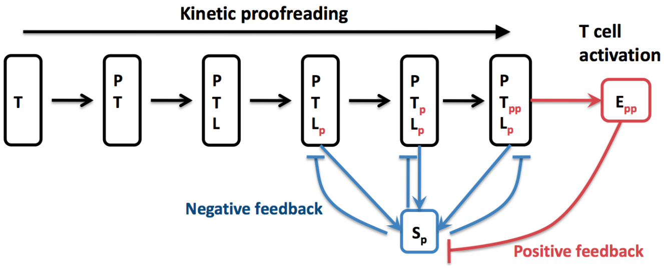


Figure 4.15: Overview of the TCR Signaling Model of Ref. [LHFH08].

antigen-presenting cell—while not responding to the large quantities of endogenous (host) pMHC expressed on all cells. The T cell appears to maintain this delicate balancie between sensitivity and selectivity through a combination of mechanisms that include kinetic proofreading, which discriminates against pMHC-receptor interactions that are too short, positive feedback, which amplifies the response and makes it more switch-like, and negative feedback, which acts in concert with kinetic proofreading to dampen responses to weak stimulation and with positive feedback to enhance the stability of the inactive state.

A computational model incorporating all three of these mechanisms has recently been developed by Lipniacki et al. [LHFH08], and serves as the basis for the experiments we conduct here. This model extends previous simplified models of kinetic proofreading [McK95] and feedback regulation [RBL$^+$96] by incorporating mechanistic detail about the involvement of specific signaling molecules.

This model captures three important properties of T cell activation, which are sensitivity to small numbers of pMHC with high binding affinity, high selectivity between pMHCs of different affinity, and antagonism, the inhibition of response by pMHC of intermediate affinity. Because only small numbers of high-affinity pMHC ligands are displayed on cell surfaces, stochastic effects have a major influence on the dynamics both of the model and of the initiation of signaling through the TCR. The model also exhibits bistable ERK responses over a broad range of pMHC number and binding affinity. This bistable regime has the interesting property that stochastic trajectories may exhibit completely different dynamics from the deterministic trajectory from the same initial state, and even the average behavior of stochastic trajectories may differ qualitatively from the deterministic behavior (see Fig. 7B of [LHFH08] for an example). This divergence between the stochastic and deterministic dynamics was the motivation for using this model of TCR as the basis for the current study, which aims to show that formal verification methods can be useful for the characterization of rule-based biochemical models. As shown in Fig. 4.3.2, under many input conditions traces from stochastic simulations may sample both stable steady states and thus diverge from deterministic traces starting from the same initial conditions, which sample only a single steady state.

We analyzed the T Cell Receptor model using the BioNetGen stochastic simulation engine, which has a CTMC semantics. We chose the Bayes Factor threshold for our experiments to be 100. We were interested in the truth of the hypothesis that the system can go from the inactive state to the active state. We verified the following property with various values of probability ρ.

$$Pr_{\geqslant\rho}(\mathbf{F}^{100}(ppERK/totalERK < 0.1) \wedge \mathbf{F}^{400}(ppERK/totalERK > 0.9))$$

The formula states that the ratio of ppERK to totalERK is below 0.1 within 100 time steps, and the ratio of ppERK to totalERK is above 0.9 within 400 time steps. It basically says

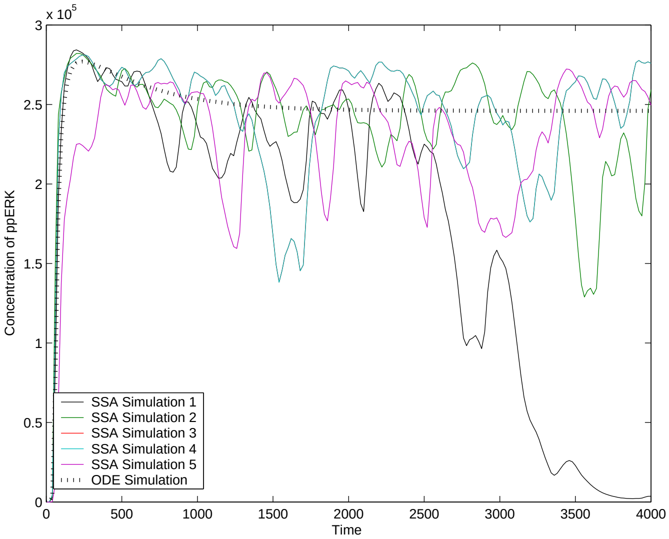


Figure 4.16: Traces from Deterministic (ODE) and Stochastic Simulation of the TCR Signaling Model. N_1 : 100, N_2 : 3000.

that the system can evolve from an inactive state to an active state within 400 time steps.

ρ	Result	Successful Samples	Failure Samples	Time
0.95	No	0	1	3s
0.75	No	0	3	10s
0.55	No	0	5	18s
0.25	No	0	16	55s

Table 4.3: Uniform Prior with Initial Number of Agonist pMHC : 100, Initial Number of Antagonist pMHC : 1000 , Bayes Factor Threshold: 100

In our first experiment, there were 100 molecules of agonist pMHC and 1000 molecules of antagonist pMHC. The results are presented in Table 4.3. As expected, the number of samples needed to decide the property depends both upon the fraction of samples that satisfied the property and the probability with which we want the property to be satisfied. If

ρ	Result	Successful Samples	Failure Samples	Time
0.95	No	41	6	4m2s
0.75	Yes	56	8	5m19s
0.55	Yes	10	1	57s
0.25	Yes	6	3	44s
0.15	Yes	1	0	5s

Table 4.4: Uniform Prior with Initial Number of Agonist pMHC : 100, Initial Number of Antagonist pMHC : 0 , Bayes Factor Threshold: 100

the two probabilities are close together, we need a large number of samples. On the other hand, we may need as few as 2 samples if these two probabilities are far apart.

Our second model started with 100 molecules of agonist pMHC (with dissociation constant 1/20 per second) while antagonist pMHC was assumed to be absent in the initial state. We also set the dissociation constant of agonist pMHC as 1/20 per second and that of antagonist pMHC as 1/3. The results are illustrated in Table 4.4. Once again, we see that the Bayesian algorithm needs to generate only a few samples if the probability of the formula being true on the model and the probability threshold in the formula are very far apart. On the other hand, if the probability of the formula being true and the probability threshold in the formula are close together, the number of samples needed increases.

The second model showed a *qualitative* difference in behavior from the first. We verified the property that the stochastic model of the T Cell Receptor pathway can go from the inactive to the active state with at least 0.75 probability in the second model. However, the probability of this transition is at less than 0.25 in the first model.

4.4 Algorithm 2: Cost Based Statistical Bayesian Model Checking

Most of the existing work in Statistical Model Checking is equipped with Frequentist guarantees of Type I and II errors. One of the key advantages of a Bayesian approach is that it can easily incorporate other decision criteria, such as the expected monetary costs for generating samples and for returning an incorrect answer. In this section, we modify Algorithm 1 and convert it into a cost-based algorithm. That is, one where we know the cost of drawing samples from the model and the cost of making an incorrect decision. Our algorithm will stop when the cost of making an additional observation is not offset by the reduction in the expected loss from making a wrong decision [Gho, Ber93]. This is the usual Bayesian way of managing the risk associated with making a wrong decision.

Consider the Model Checking query that a newly designed electronic braking system is correct with 99.999999% probability. Each detailed simulation of such an event takes a few days and is associated with a non-trivial computational cost. Each day of simulation time of a single CPU node costs about $20 on the Amazon EC2 cluster. The cost of making an incorrect decision is substantial. If an erratic model is released into the market, the cost of a recall is prohibitive. In 2010, the cost of an automotive recall in the US was estimated to be over $2 billion. The goal of the cost based Statistical Model Checking approach is to use the information about the cost of sampling and the cost of making an incorrect decision to decide the number of samples that should be observed.

We now suggest a Bayesian approach to deciding the Statistical Model Checking problem when the cost of generating each simulation and the cost of making a wrong decision is known. When n independent simulation traces of the system have been analyzed against an *AFM* formula ϕ and exactly x of these traces indeed satisfy the specification, then the posterior distribution of the probability with which $\mathcal{M}$ satisfies the *AFM* formula ϕ is given by the product of the prior probability and the probability of the observed samples. After having made n observations, the probability that the system $\mathcal{M}$ satisfies the *AFM* formula ϕ with probability at least ρ is

$$\frac{\displaystyle\int_0^{\rho} f(x_1|u) \cdots f(x_n|u) \cdot g(u)\, du}{\displaystyle\int_0^{1} f(x_1|u) \cdots f(x_n|u) \cdot g(u)\, du}$$

If C_0 is the possible cost of the hazard associated with rejecting the property $Pr_{\geqslant\rho}(\phi)$ when the property is actually true, the expected cost Cost $(N = n)$ associated with making a wrong decision after N samples is then given by

$$\text{Cost } (N = n\,) \quad = \quad \frac{\displaystyle\int_0^{\rho} f(x_1|u) \cdots f(x_n|u) \cdot g(u)\, du}{\displaystyle\int_0^{1} f(x_1|u) \cdots f(x_n|u) \cdot g(u)\, du}\ C_0 \tag{4.14}$$

If C_1 is the possible cost of the hazard associated with accepting the property $Pr_{\geqslant\rho}(\phi)$ when the property is actually false, the expected cost Cost $(N = n)$ associated with making a wrong decision after N samples is then given by

$$\text{Cost } (N = n\,) \;=\; \frac{\displaystyle\int_{\rho}^{1} f(x_1|u)\cdots f(x_n|u)\cdot g(u)\; du}{\displaystyle\int_{0}^{1} f(x_1|u)\cdots f(x_n|u)\cdot g(u)\; du}\; C_1 \tag{4.15}$$

The Cost based Bayesian Model Checking algorithm is illustrated in Algorithm 1. The algorithm takes six inputs:

(i) the model $\mathcal{M}$ under investigation,

(ii) the *Probabilistic Adapted Finitely Monitorable (PAFM)* Formula $Pr_{\geqslant\rho}(\phi)$,

(iii) the cost of generating each sample s,

(iv) the costs C_0 and C_1 of the possible hazard associated with the failure modes of the property,

(v) the indifference region $[\rho - \epsilon_1, \rho + \epsilon_2]$ such that $0 \leqslant \rho - \epsilon_1 < \rho < \rho + \epsilon_2 \leqslant 1]$ such that we are indifferent to the answer of the Statistical Model Checking algorithm if the true probability of satisfying the formula ϕ lies within the tolerance region, and

(vi) the prior probability distribution g for the probability of the model $\mathcal{M}$ satisfying the formula ϕ.

The algorithm then performs i.i.d simulations of the model under investigation and records the total number of simulation so far performed (n) and the number of simulations that actually satisfy the AFM formula (x). After observing each sample, the algorithm verifies whether the cost of observing another additional sample exceeds the reduction in the expected loss from an incorrect decision by the Model Checking algorithm (irrespective of the outcome of this additional sample). If an additional sample is less expensive to generate than the associated reduction in the expected loss from an incorrect decision by the Model Checking algorithm, the algorithm loops back and generates a new sample. Otherwise, the algorithm stops and generates the answer to the model checking query of the system $\mathcal{M}$ satisfying the *Probabilistic Adapted Finitely Monitorable (PAFM)* formula $Pr_{\geqslant \rho}(\phi)$ depending on the value of the Bayes Factor after observing these samples.

4.4.1 Theorems

Theorem 7 *The Cost based Bayesian Statistical Model Checking algorithm terminates almost surely if the true probability with which the model $\mathcal{M}$ satisfies the formula ϕ lies in the KL support of the proper prior probability distribution g.*

Proof 7 *There are two cases:*

(i) Suppose the PAFM *specification is true i.e. $\rho_0 > \rho + \epsilon_2$, then the interval $[0, \rho + \epsilon_2]$ is strongly δ-separated from ρ_0 for some non-zero constant δ (See Example 10). Then, we know that*

$$P\left(\frac{\displaystyle\int_0^{\rho+\epsilon_2} f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du}{\displaystyle\int_0^1 f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du} \geqslant e^{-nb}\ i.o.\right) = 0 \tag{4.16}$$

$$\implies \frac{\displaystyle\int_0^{\rho+\epsilon_2} f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du}{\displaystyle\int_0^1 f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du} \leqslant e^{-nb}\ i.o.\ \ a.s. \tag{4.17}$$

Algorithm 2 Cost based Bayesian Model Checking algorithm

```
Require: Model $\mathcal{M}$, PAFM Formula $Pr_{\geqslant\rho}(\phi)$ , Cost of each simulation $s$, the costs
  $C_0$ and $C_1$ of the possible hazard associated with the failure modes of the property,
  Indifference Region $[\rho - \epsilon_1, \rho + \epsilon_2]$, Prior density $g$ for unknown parameter $u$

  $n \leftarrow 0$; /* Total Number of Samples observed so far */
  $i \leftarrow 0$; /* Samples satisfying the AFM formula */
  $ExpectedLoss = \infty$;

  while $ExpectedLoss > s$ do
    $n \leftarrow n + 1$;
    Observe an i.i.d. sample simulation $\sigma_n$;
    if $\sigma_n \models \phi$ then
      $x_n = 1, \quad i \leftarrow i + 1$; /*simulation satisfies the AFM formula */
    else
      $x_n = 0$; /*simulation does not satisfy the AFM formula */
    end if
    $ExpectedLoss = min\left(\frac{\int_0^{\rho-\epsilon_1} f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du}{\int_0^1 f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du}\ C_0, \frac{\int_{\rho+\epsilon_2}^1 f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du}{\int_0^1 f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du}\ C_1\right)$
  end while
  if $\left(\frac{\int_0^{\rho-\epsilon_1} f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du}{\int_0^1 f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du}\ C_0\ <\ \frac{\int_{\rho+\epsilon_2}^1 f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du}{\int_0^1 f(x_1|u)\cdots f(x_n|u)\cdot g(u)\,du}\ C_1\right)$
  then
    print The formula $Pr_{\geqslant\rho}(\phi)$ holds on $\mathcal{M}$.
  else
    print The formula $Pr_{\geqslant\rho}(\phi)$ does not hold on $\mathcal{M}$.
  end if
```

$$\Longrightarrow \frac{\displaystyle\int_0^{\rho-\epsilon_1} f(x_1|u)\cdots f(x_n|u)\cdot g(u)\;du}{\displaystyle\int_0^1 f(x_1|u)\cdots f(x_n|u)\cdot g(u)\;du} \leqslant e^{-nb}\; i.o.\;\; a.s. \quad (\textit{Probability densities are positive}) \tag{4.18}$$

$$\Longrightarrow \frac{\displaystyle\int_0^{\rho-\epsilon_1} f(x_1|u)\cdots f(x_n|u)\cdot g(u)\;du}{\displaystyle\int_0^1 f(x_1|u)\cdots f(x_n|u)\cdot g(u)\;du} C_0 \leqslant e^{-nb} C_0\;\; i.o.\;\; a.s. \tag{4.19}$$

(ii) Suppose the PAFM *specification is false i.e. $\rho_0 < \rho - \epsilon_1$, then the interval $[\rho - \epsilon_1, 1]$ is strongly δ-separated from ρ_0 for some non-zero constant δ (See Example 10).*

Then, we know that

$$P\left(\frac{\displaystyle\int_{\rho-\epsilon_1}^1 f(x_1|u)\cdots f(x_n|u)\cdot g(u)\;du}{\displaystyle\int_0^1 f(x_1|u)\cdots f(x_n|u)\cdot g(u)\;du} \geqslant e^{-nb}\; i.o.\right) = 0 \tag{4.20}$$

$$\Longrightarrow \frac{\displaystyle\int_{\rho-\epsilon_1}^1 f(x_1|u)\cdots f(x_n|u)\cdot g(u)\;du}{\displaystyle\int_0^1 f(x_1|u)\cdots f(x_n|u)\cdot g(u)\;du} \leqslant e^{-nb}\; i.o.\;\; a.s. \tag{4.21}$$

$$\Longrightarrow \frac{\displaystyle\int_{\rho+\epsilon_2}^1 f(x_1|u)\cdots f(x_n|u)\cdot g(u)\;du}{\displaystyle\int_0^1 f(x_1|u)\cdots f(x_n|u)\cdot g(u)\;du} \leqslant e^{-nb}\; i.o.\;\; a.s. \quad (\textit{Probability densities are positive}) \tag{4.22}$$

$$\Longrightarrow \frac{\displaystyle\int_{\rho+\epsilon_2}^1 f(x_1|u)\cdots f(x_n|u)\cdot g(u)\;du}{\displaystyle\int_0^1 f(x_1|u)\cdots f(x_n|u)\cdot g(u)\;du} C_1 \leqslant e^{-nb} C_1\;\; i.o.\;\; a.s. \tag{4.23}$$

In both cases, the computed expected loss vanished to zero as we increase the number of samples obtained. Thus, it will become strictly smaller that the cost of generating each sample s and the algorithm stops.

Cost of Simulation much smaller than the Loss associated with an Erratic decision

An interesting scenario happens when the cost of generating each sample observation through simulation for the model i.e. s is much smaller than the loss associated with an error C i.e. $\frac{s}{C} \rightarrow 0$. The number of samples needed before the algorithm terminates is bounded by the logarithm of the ratio $\frac{C}{s}$. Hence, even though the ratio $\frac{C}{s}$ itself may be very large, the number of samples needed for the algorithm to terminate only grows logarithmically in this ratio.

Cost of Simulation close to the Loss associated with an Erratic decision

When the the cost of generating each sample observation through simulation for the model i.e. s is close to the loss associated with an error C i.e. $\frac{s}{C} \rightarrow 1$, then the number of samples needed by our algorithm is approximately a linear function of the ratio $\frac{C}{s}$ itself. This can be demonstrated by using the observation that the Taylor series expansion for the term $\log x$ is approximately $x - 1$, when x is close to 1 and the higher order nonlinear terms are small enough to be ignored.

Cost of Simulation larger than the Loss associated with an Erratic decision

Another interesting scenario happens when the cost of generating each sample observation through simulation for the model i.e. s is much larger than the loss associated with an error C i.e. $\frac{C}{s} < 1$. In this case, it is clear that even drawing one sample the system is going to be more expensive that the cost of an error made by the Cost based Bayesian Statistical Model Checking algorithm. In this case, it is not useful to validate this model using simulation based sampling.

4.4.2 Empirical Performance of Algorithm 2

We analyzed the performance of the algorithm on the T Cell Receptor pathway [LHFH08]. From experimental observations, we know that it takes at least 4 seconds to run a simulation of this model on a 2.2 GHz Quad-Core AMD Opteron Processor. We also know that the cost of running the simulation on an Amazon EC2 cluster for an hour is 8.5 cents. Thus, the cost of generating a single sample for this model s is at least 9×10^{-5} cents.

The loss from the algorithm producing an incorrect answer depends on the context of the application in which this model checking problem arises. For example, if the incorrect answer could lead to a product recall that costs $\$1$ million, that would be the cost of producing an incorrect answer. We study the number of samples required by the algorithm as a function of the loss from the incorrect answer.

We analyzed a property that should hold on the model with at least probability 0.95. We used uniform priors and no indifference region. We also know that this property is true on the model with probability 0.88. We varied the cost of making an incorrect decision but kept the losses due to the algorithm producing an incorrect answer symmetric i.e. $C_0 = C_1 = C(say)$.

Cost of Incorrect Decision (C)	#Samples Needed	Expected Loss at Termination
0.01	68	0.000073
0.1	127	0.000069
1	150	0.000071
10	158	0.000049
100	184	0.000065
1,000	241	0.000065
10,000	253	0.000067
100,000	273	0.000047
1,000,000	287	0.000056

Table 4.5: Performance of Cost based Bayesian Model Checking Algorithm - I

For the same system and the same property, we sought an answer to the question that the property is true with probability at least 0.5. Again, we used uniform priors and no indifference region for our algorithm. We varied the cost of making an incoorect decision but kept the costs symmetric.

In Table 4.5 and Table 4.6, we present the number of samples needed by the Bayesian Statistical Model Checking algorithm. We observe that the number of samples needed increases as we increase the cost of making a wrong decision.

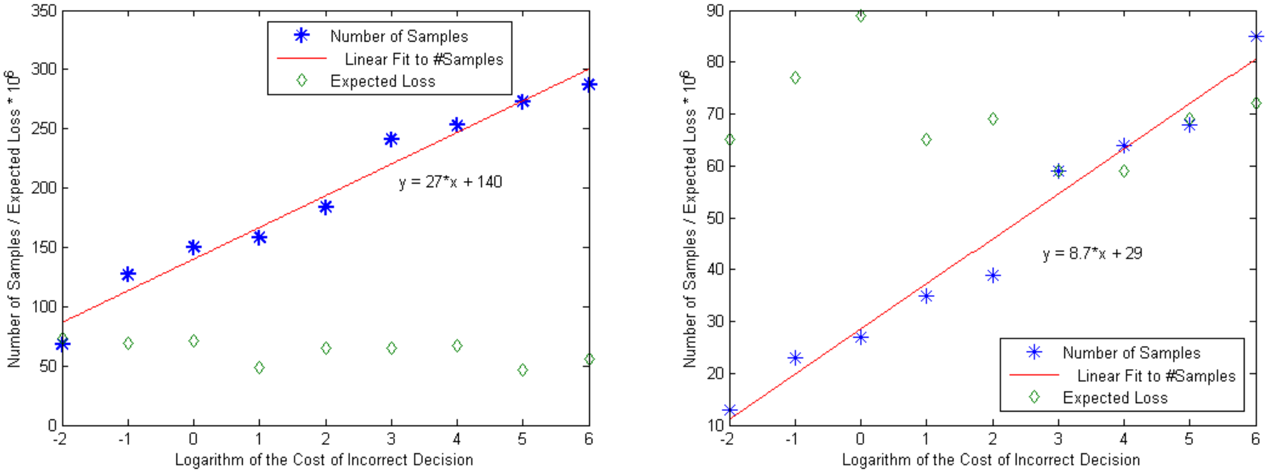
In Figure 4.17(a) and Figure 4.17(b), we plotted the number of samples observed before our algorithm terminates against the logarithm of the symmetric cost of making an incorrect decision. An important point to note is that the number of samples only increases logarithmically in the cost of an incorrect decision. This is explained by the fact

Cost of Incorrect Decision (C)	#Samples Needed	Expected Loss at Termination
0.01	13	0.000065
0.1	23	0.000077
1	27	0.000089
10	35	0.000065
100	39	0.000069
1,000	59	0.000059
10,000	64	0.000059
100,000	68	0.000069
1,000,000	85	0.000072

Table 4.6: Performance of Cost based Bayesian Model Checking Algorithm - II

that Bayesian posteriors concentrate exponentially as discussed in Theorems 2 and 8. We also plot the expected loss from making an incorrect decision at the termination of the algorithm. We see that the expected loss has fallen below the cost of a simulation when the algorithm terminates.

The exponential decay in expected loss from an incorrect decision merits further discussion from the perspective of practical applications of this algorithm. If our upper estimate of making a wrong decision is off by a constant factor, the algorithm would only need to draw constant many additional samples. Similarly, if our lower estimate on the cost of drawing a single observation using simulation is off by a constant factor, the cost based Bayesian Statistical Model Checking algorithm only needs to draw constant many additional samples. While it is probably impossible to precisely compute the cost of each simulation or the cost of making a wrong decision in advance, it is reasonable to assume that good lower bounds on the cost of simulating the model and upper bounds on the cost of making a wrong decision are readily available. The performance of our algorithm is



(a) Actual Probability = 0.88

(b) Actual Probability = 0.5

Figure 4.17: Performance as a Function of Cost of Incorrect Decision

not severely affected by the use of these approximate costs for simulation and making an incorrect decision because of the exponential concentration of the Bayesian posterior in our setting.

4.5 Algorithm 3: Non-i.i.d. Bayesian Statistical Model Checking

Both of the Bayesian statistical model checking algorithms presented earlier in this chapter use independent and identically distributed samples to explore the stochastic model. Indeed, to the best of our knowledge, all existing statistical model checking algorithms [CFL$^+$08, JJ08, JCL$^+$09, YS02, YKNP04, YKNP06, You05a, You05c, You05b, You04, GS05, SVA04,

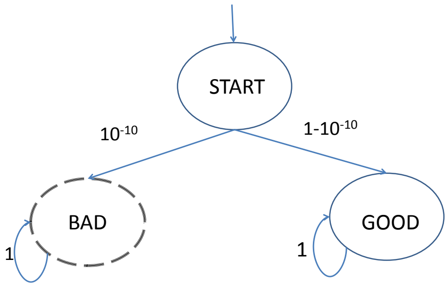


Figure 4.18: Statitisical Verification using i.i.d. Samples would need about 10^{10} Samples to observe the Erratic Behavior.

SVA05] sample from the model in an i.i.d. fashion. That is, each sample trace is drawn from the *same* probability distribution, and all traces are mutually independent. In contrast, our algorithm presented in this section draws sample traces in a non-identically distributed manner, but ensures that all sample traces are mutually independent. We refer to this as non-i.i.d. sampling.

The distinction between i.i.d. and non-i.i.d. sampling is important. In particular, the error bounds reported by existing statistical model checking algorithms are calculated under the assumption that the samples are generated in an i.i.d. fashion. That is, each trace is generated independently and according to the same underlying probability distribution. Unfortunately, i.i.d. sampling is a very poor way of investigating rare events. To see why, consider Figure 4.18. A visual inspection of the model illustrates that the bad state is reachable with probability 10^{-10}. Thus, the expected number of i.i.d. samples needed to produce one trace that visits the bad state is 10^{10}. This highlights one of the fundamental limitations of existing statistical model checking algorithms — rare events are unlikely to be sampled. To address this problem, we develop a new approach to statistical model checking based on non-i.i.d. sampling that can expose rare behaviors and yet still provide statistical bounds on the probability of an incorrect answer.

The key idea behind our non-i.i.d. sampling strategy based algorithm is that we do not treat the model as a black box. Instead, we allow the model to be explicitly but carefully manipulated before generating each sample. Each individual perturbation changes the measure on the sampled traces, but we ensure that combination of all perturbations is such that the ratio between the true and a suitable defined "average" of perturbed measures is bounded. This enables us to provide statistical bounds on the probability of an incorrect answer.

4.5.1 Background

In this section, we will discuss the conditions under which (a) it is possible to bound the error introduced by our change of measures, as in our non-i.i.d. sampling procedure, and (b) that a Bayesian Sequential Hypothesis Testing based procedure is guaranteed to terminate, almost surely even under the non- i.i.d. scenario.

A stochastic model $\mathcal{M}$ is naturally associated with a probability measure μ. Rather than drawing samples σ_i from $\mathcal{M}$ under μ (which is what an i.i.d. sampling strategy would do), our sampling strategy can be thought of as the assignment of a set of probability measures $\mu_1, \mu_2, ...$ to $\mathcal{M}$. Each unique sample σ_i is associated with an *implied* probability measure μ_i and is generated from $\mathcal{M}$ under μ_i in an i.i.d. manner. Our proofs require that all the implied probability measures are *equivalent*. That is, an event is possible (resp. impossible) under a probability measure *if and only if* it is possible (resp. impossible) under the original probability measure μ.

There are two key technical challenges associated with our non-i.i.d. approach. The first is constructing the implied probability measures for stochastic models. For stochastic differential equation models, this is addressed using Girsanov's theorem [Gir60]. For models written in the probabilistic reactive module formalism, we will suggest a method to generate the non-i.i.d. samples and their implied probability measures in Section. 4.5.4. The second challenge is computing the Type-I/Type-II errors under a non-i.i.d. sampling procedure. This is the subject of our discussion in this section.

Our method is related to importance sampling - a general technique for estimating properties of one probability distribution, while only having samples generated from a different probability distribution rather than the distribution of interest. In conjunction with Monte Carlo estimation, the idea behind importance sampling is that certain values of the input random variables have more impact on the parameter being estimated than others. The basic approach in importance sampling is to choose a probability distribution which favours the important values. Clearly, a direct use of biased distributions will result in a biased estimator if applied directly. However, if the simulation outputs are weighted to correct for the bias, and this ensures that the importance sampling based estimator is unbiased.

The work presented here is different from importance sampling as we do not know the probability distribution from which we wanted to sample. Moreover, we do not sample from one fixed distribution but a number of biased probability distributions. Further, we do not need the resulting samples to be completely free of bias but permit a small bias in the samples we draw. While our work is closely related in spirit to the idea of importance sampling in that we also employ the general technique of change of measures, the similarities end there.

Bounding Errors Under a Change of Measure

In this section, we develop the machinery needed to compute bounds on the Type-I/Type-II errors for a testing strategy based on non-i.i.d. samples. In classical statistical model checking, the samples are drawn from the probability space with a known measure μ in an i.i.d. fashion and then Wald's SPRT or Bayesian posterior computation is used to derive bounds on the Type-I/II errors. However, this is no longer sufficient when the samples are drawn in an *independent but non-identically distributed* fashion. Suppose a given behavior, say ϕ, holds on the original model with an (unknown) probability ρ_0.

$$P(\rho_0 < \rho | X_i) = \frac{\displaystyle\int_0^\rho p_\mu(X_i|u)\ g(u)\ du}{\displaystyle\int_0^1 p_\mu(X_i|u)\ g(u)\ du} \tag{4.24}$$

Here, X_i is a Bernoulli random variable denoting the event that i^{th} sample satisfies the given behavior ϕ. Note that the X_is must be independent of one another. Now, we can rewrite the above expression as:

$$P(\rho_0 < \rho | X_i) = \frac{\displaystyle\int_0^\rho \frac{p_\mu(X_i|u)}{p_{\mu_i}(X_i|u)} p_{\mu_i}(X_i|u)\ g(u) du}{\displaystyle\int_0^1 \frac{p_\mu(X_i|u)}{p_{\mu_i}(X_i|u)} p_{\mu_i}(X_i|u)\ g(u)\ du} \tag{4.25}$$

Note that the term $p_{\mu_i}(X_i|u)$ denotes the probability of observing the event X_i under the probability measure μ_i if the unknown probability ρ were u. We call the ratio $\dfrac{p_{\mu_i}(X_i|u)}{p_\mu(X_i|u)}$ the *implied Radon-Nikodym derivative* for the change of measure between the two equivalent probability measures that we have constructed. Suppose, the testing strategy has made n observations $X_1, X_2, \ldots X_n$. Then,

$$P(\rho_0 < \rho|X) = \frac{\displaystyle\int_0^\rho \prod_{i=1}^{n} \left(\left(\frac{p_\mu(X_i|u)}{p_{\mu_i}(X_i|u)} \right) p_{\mu_i}(X_i|u) \right) g(u)du}{\displaystyle\int_0^1 \prod_{i=1}^{n} \left(\left(\frac{p_\mu(X_i|u)}{p_{\mu_i}(X_i|u)} \right) p_{\mu_i}(X_i|u) \right) g(u)du} \tag{4.26}$$

Note that we cannot easily compute the change of measure $\frac{p_{\mu_i}(X_i|u)}{p_\mu(X_i|u)}$ algebraically or numerically. Our algorithm does not need to compute this quantity explicitly but only lower and upper bounds on the change of measure. Note that $\frac{p_{\mu_i}(X_i|u)}{p_\mu(X_i|u)}$ is not the change of measure for the probability of a given observed sample; rather, it is the change of probability of the AFM specification ϕ being true on the model. In other words, $\frac{p_{\mu_i}(X_i|u)}{p_\mu(X_i|u)}$ is the change in probability of any *random* sample drawn from the model satisfying the formula ϕ, and is not related to the probability of observing a given particular randomly drawn sample under the original and the new probability measure. In order to compute this change of measure $\frac{p_{\mu_i}(X_i|u)}{p_\mu(X_i|u)}$ explicitly, one would have to enumerate all the paths that satisfy the specification ϕ. If the model were small enough to do such an enumeration of paths, there would be no need to apply statistical model checking algorithm to such a model. Hence, our algorithm avoids computing the change of measure explicitly.

Consider the following expression that is computable without knowing the implied

Radon-Nikodym derivative or change of measure explicitly. We simply draw an i.i.d. sample under the probability measure μ_i from the model $\mathcal{M}$ and then compute the integrals.

$$Q(\rho_0 < \rho | X_i) = \frac{\displaystyle\int_0^\rho p_{\mu_i}(X_i|u)\ g(u)\ du}{\displaystyle\int_0^1 p_{\mu_i}(X_i|u)\ g(u)\ du} \tag{4.27}$$

Now, we can rewrite the above expression as:

$$Q(\rho_0 < \rho | X_i) = \frac{\displaystyle\int_0^\rho \left(\frac{p_{\mu_i}(X_i|u)}{p_\mu(X_i|u)}\right) p_\mu(X_i|u)\ g(u)\ du}{\displaystyle\int_0^1 \left(\frac{p_{\mu_i}(X_i|u)}{p_\mu(X_i|u)}\right) p_\mu(X_i|u)\ g(u)\ du} \tag{4.28}$$

Our result will exploit the fact that we do not allow our testing or sampling procedures to have arbitrary implied Radon-Nikodym derivatives. This is reasonable as no statistical guarantees should be available for an intelligently designed but adversarial test procedure that (say) tries to avoid sampling from the given behavior. Suppose that the implied Radon-Nikodym derivative $\frac{p_{\mu_i}(X_i|u)}{p_\mu(X_i|u)}$ always lies between a constant c and another constant $1/c$. That is, the change of measure does not distort the probabilities of observable events by more than a factor of c. Then, we observe that:

$$Q(\rho_0 < \rho | X_i) = \frac{\displaystyle\int_0^\rho \frac{p_{\mu_i}(X_i|u)}{p_\mu(X_i|u)} p_\mu(X_i|u) g(u)\ du}{\displaystyle\int_0^1 \frac{p_{\mu_i}(X_i|u)}{p_\mu(X_i|u)} p_\mu(X_i|u) g(u)\ du} \tag{4.29}$$

$$\leqslant \frac{\displaystyle\int_0^\rho c p_\mu(X_i|u) g(u)\ du}{\displaystyle\int_0^1 \frac{1}{c} p_\mu(X_i|u) g(u)\ du} \tag{4.30}$$

$$= \quad c^2 \, P(\rho_0 < \rho | X_i) \tag{4.31}$$

Furthermore,

$$Q(\rho_0 < \rho | X_i) = \frac{\displaystyle\int_0^{\rho} \frac{p_{\mu_i}(X_i|u)}{p_\mu(X_i|u)} p_\mu(X_i|u) g(u) \, du}{\displaystyle\int_0^{1} \frac{p_{\mu_i}(X_i|u)}{p_\mu(X_i|u)} p_\mu(X_i|u) g(u) \, du} \tag{4.32}$$

$$\geqslant \frac{\displaystyle\int_0^{\rho} \frac{1}{c} p_\mu(X_i|u) g(u) \, du}{\displaystyle\int_0^{1} c p_\mu(X_i|u) g(u) \, du} \tag{4.33}$$

$$= \frac{1}{c^2} \, P(\rho_0 < \rho | X_i) \tag{4.34}$$

Thus, by allowing the sampling algorithm to change measures by at most c, we have changed the posterior probability of observing a behavior given a single sample by at most c^2. Suppose, the testing strategy has made n observations $X_1, X_2, \ldots X_n$. Then,

$$Q(\rho_0 < \rho | X) = \frac{\displaystyle\int_0^{\rho} \prod_{i=1}^{n} \left(\frac{p_{\mu_i}(X_i|u)}{p_\mu(X_i|u)} p_\mu(X_i|u) \right) g(u) \, du}{\displaystyle\int_0^{1} \prod_{i=1}^{n} \left(\frac{p_{\mu_i}(X_i|u)}{p_\mu(X_i|u)} p_\mu(X_i|u) \right) g(u) \, du} \tag{4.35}$$

$$\leqslant \frac{\displaystyle\int_0^{\rho} c^n \prod_{i=1}^{n} \left(p_\mu(X_i|u) \right) g(u) \, du}{\displaystyle\int_0^{1} \frac{1}{c^n} \prod_{i=1}^{n} \left(p_\mu(X_i|u) \right) g(u) \, du} \tag{4.36}$$

$$= \quad c^{2n} \, P(\rho_0 < \rho | X_1, X_2, \ldots X_n) \tag{4.37}$$

Similarly,

$$Q(\rho_0 < \rho | X_1, \ldots X_n) \geqslant \frac{1}{c^{2n}} \frac{\displaystyle\int_0^{\rho} \prod_{i=1}^{n} (p_\mu(X_i|u))\, g(u)\, du}{\displaystyle\int_0^{1} \prod_{i=1}^{n} (p_\mu(X_i|u))\, g(u)\, du} \tag{4.38}$$

$$= \frac{1}{c^{2n}} P(\rho_0 < \rho | X_1, \ldots X_n) \tag{4.39}$$

Hence, we have changed the posterior probability of observing a behavior ϕ given n samples by at most c^{2n} by permitting implied Radon-Nikodym derivatives of at most c.

Termination Conditions for non-i.i.d. Sampling

In Theorem 8, we have shown that a Bayesian Sequential Hypothesis testing procedure with i.i.d. sampling will terminate almost surely. However, our algorithm 4.5.2 uses non-i.i.d. samples and so we must consider the conditions under which such an algorithm will terminate. First, note that the factor introduced due to the change of measure c^{2n} in Equations 4.37 and 4.39 can easily outweigh the gain made by the exponential concentration of the posterior probability measure $P(\rho_0 < \rho | X_1, X_2, \ldots X_n)$ (See Theorem 2). This is not surprising because our construction thus far does not force a sampling (testing) strategy *not* to bias against a sample in an intelligent way. That is, a maliciously designed sampling (testing) procedure could simply avoid the error prone regions of the design. To address this, we define the notion of a *fair* testing strategy that does not engage in such malicious sampling.

Definition 16 *A testing strategy is η-**fair** ($\eta \geqslant 1$) if and only if the geometric average of the implied* Radon-Nikodym derivatives *over a number of samples is within a constant factor η of unity, i.e.,*

$$\frac{1}{\eta} \leqslant \sqrt[n]{\prod_{i=1}^{n} \frac{p_{\mu_i}(X_i|u)}{p_{\mu}(X_i|u)}} \leqslant \eta$$

Note that a fair test strategy does *not* need to sample from the underlying distribution in an i.i.d. manner. However, it *must* guarantee that the probability of observing the given behavior in a large number of observations is not altered *substantially* by the non-i.i.d. sampling. Intuitively, we want to make sure that we bias *for* each sample as many times as we bias *against* it. Our main result shows that such a long term neutrality is sufficient to generate statistical guarantees on an otherwise non-i.i.d. testing procedure.

Definition 17 *An η-**fair** test is said to be **eventually fair** if and only if $1 \leqslant \eta^4 < e^b$, where b is the constant in the (exponential) Bayes posterior concentration theorem (See Theorem 2).*

The notion of a *eventually fair* test corresponds to a testing strategy that is not malicious or adversarial, and is making an honest attempt to sample from all the events in the long run. The testing strategy may bias in favor of the low probability events to draw some test samples, but it would also bias against such an event to draw other test samples.

4.5.2 Algorithm

We now present our Statistical Verification algorithm in terms of an *eventually fair* non-i.i.d. testing procedure sampling with “implied" change of measures. Our algorithm is relatively simple and generalizes our previous Bayesian Statistical Model Checking algorithm [JCL$^+$09] to non-i.i.d. samples using change of measures.

The algorithm draws non-i.i.d. samples from the stochastic models under different probability measures chosen by an eventually fair testing strategy. The eventually fair testing strategy ensures that the average of the implied change of measure is appropriately bounded so as to make the non-i.i.d. sampling approach converge to the same answer as the i.i.d. sampling approach. The variable n denotes the number of samples obtained so far and x denoted the number of samples that satisfy the AFM specification ϕ. Based upon the samples observed, we compute the Bayes Factor using samples obtained under the new probability measures. We know from Equations 4.37 and 4.39 that the Bayes Factor so computed is within a factor of the original Bayes Factor under the natural probability measure. Hence, the algorithm divides the Bayes Factor computed using samples obtained under the new probability measures by the factor η^{2n} if this computed Bayes Factor is larger than one. If the Bayes Factor computed using samples obtained under the new probability measures is less than one, the algorithm multiplies the computed Bayes Factor by the factor η^{2n}.

Algorithm 3 *Non- i.i.d.* Statistical Verification algorithm

```
Require: Stochastic Model $\mathcal{M}$, PAFM Specification $Pr_{\geqslant\rho}(\phi)$ , Bayes Factor Threshold
  $T > 1$, Bound on implied Radon-Nikodym derivatives $\eta$, Prior density $g$ for unknown
  parameter $u$
  Initialize $n$=0;    /*$n$ is number of samples observed*/
  Initialize $x$=0;    /*$x$ is number of successful samples*/
  $BF \leftarrow 1$
  while ( $BF > \frac{1}{T}$ and $BF < T$ ) do
    $n \leftarrow n + 1$ ;
    Choose an implied change of probability measure $c_i$ using an $\eta$-eventually fair testing
    strategy.
    Draw sample $\sigma_n$ from the stochastic model $\mathcal{M}$ under the new probability measure.
    if $\sigma_n \models \phi$ then
      $X_n \leftarrow 1$ ; $x \leftarrow x + 1$
    else
      $X_n \leftarrow 0$ ;
    end if
    $BF \leftarrow \dfrac{\int_{\rho}^{1} \prod_{i=1}^{n} p_{\mu_i}(X_i|u) g(u)\, du}{\int_{0}^{\rho} \prod_{i=1}^{n} p_{\mu_i}(X_i|u) g(u)\, du}$
    if $BF > 1$ then
      $BF \leftarrow \frac{1}{\eta^{2n}} BF$
    else
      $BF \leftarrow \eta^{2n} BF$
    end if
  end while
```

4.5.3 Results on the Non-i.i.d. Statistical Verification algorithm

The key technical accomplishment in this section is the analysis of our non-i.i.d. statistical verification algorithm. We prove the following general theorem that characterizes the priors under which our algorithm terminates almost surely:

Theorem 8 (Termination) *Our non-i.i.d. Bayesian Statistical Model Checking algorithm for the PAFM specification $Pr_{\geqslant \rho}(\phi)$ using eventually fair sampling strategies terminates almost surely if the true probability ρ_0 with which the model $\mathcal{M}$ satisfies the formula ϕ lies in the KL support of the proper prior probability distribution g and $\rho_0 \neq \rho$.*

Proof 8 *There are two cases:*

(i) Suppose the PAFM *specification is true i.e. $\rho > \rho_0$, then the interval $[0, \rho]$ is strongly δ-separated from ρ_0 for some non-zero constant δ (See Example 10). Then, we know that*

$$P\left(\frac{\displaystyle\int_0^{\rho} f_\mu(x_1|u)\cdots f_\mu(x_n|u)\cdot g(u)\,du}{\displaystyle\int_0^{1} f_\mu(x_1|u)\cdots f_\mu(x_n|u)\cdot g(u)\,du} \geqslant e^{-nb}\ i.o.\right) = 0 \tag{4.40}$$

$$\implies \frac{1}{\eta^{2n}}\frac{\displaystyle\int_0^{\rho} f_{\mu_1}(x_1|u)\cdots f_{\mu_n}(x_n|u)\cdot g(u)\,du}{\displaystyle\int_0^{1} f_{\mu_1}(x_1|u)\cdots f_{\mu_n}(x_n|u)\cdot g(u)\,du} \leqslant e^{-nb}\ i.o.\ \ a.s.\ \ (\textit{From Equation 4.37})$$

$$\implies \frac{\displaystyle\int_0^\rho f_{\mu_1}(x_1|u)\cdots f_{\mu_n}(x_n|u)\cdot g(u)\;du}{\displaystyle\int_0^1 f_{\mu_1}(x_1|u)\cdots f_{\mu_n}(x_n|u)\cdot g(u)\;du} \leqslant \eta^{2n}e^{-nb}\;i.o.\;\;a.s.\;\;(\textit{Algebraic Manipulation}) \tag{4.41}$$

$$\implies \frac{\displaystyle\int_\rho^1 f_{\mu_1}(x_1|u)\cdots f_{\mu_n}(x_n|u)\cdot g(u)\;du}{\displaystyle\int_0^1 f_{\mu_1}(x_1|u)\cdots f_{\mu_n}(x_n|u)\cdot g(u)\;du} \geqslant 1-\eta^{2n}e^{-nb}\;i.o.\;\;a.s.\;\;(\textit{Posterior integrates to 1}) \tag{4.42}$$

$$\begin{aligned}\implies \frac{1}{\eta^{2n}}\frac{\displaystyle\int_\rho^1 f_{\mu_1}(x_1|u)\cdots f_{\mu_n}(x_n|u)\cdot g(u)\;du}{\displaystyle\int_0^\rho f_{\mu_1}(x_1|u)\cdots f_{\mu_n}(x_n|u)\cdot g(u)\;du} &\geqslant \frac{1}{\eta^{2n}}\frac{1-\eta^{2n}e^{-nb}}{\eta^{2n}e^{-nb}}\;\;(\textit{Using Eqns. 4.41 and 4.42})\\ &\geqslant \eta^{-4n}e^{nb}-1\;\;(\textit{Algebraic Manipulation})\end{aligned} \tag{4.43}$$

(ii) Suppose the PAFM *specification is false i.e.* $\rho < \rho_0$*, then the interval* $[\rho, 1]$ *is strongly* δ*-separated from* ρ_0 *for some non-zero constant* δ *(See Example 10). Then, we know that*

$$P\left(\frac{\displaystyle\int_\rho^1 f_\mu(x_1|u)\cdots f_\mu(x_n|u)\cdot g(u)\;du}{\displaystyle\int_0^1 f_\mu(x_1|u)\cdots f_\mu(x_n|u)\cdot g(u)\;du} \geqslant e^{-nb}\;i.o.\right) = 0 \tag{4.44}$$

$$\implies \frac{\displaystyle\int_\rho^1 \frac{1}{c_1}f_{\mu_1}(x_1|u)\cdots \frac{1}{c_n}f_{\mu_n}(x_n|u)\cdot g(u)\;du}{\displaystyle\int_0^1 \frac{1}{c_1}f_{\mu_1}(x_1|u)\cdots \frac{1}{c_n}f_{\mu_n}(x_n|u)\cdot g(u)\;du} \leqslant e^{-nb}\;i.o.\;\;a.s.\;\;(\textit{From Equation 4.37})$$

$$\Longrightarrow \frac{\displaystyle\int_\rho^1 f_{\mu_1}(x_1|u)\cdots f_{\mu_n}(x_n|u)\cdot g(u)\,du}{\displaystyle\int_0^1 f_{\mu_1}(x_1|u)\cdots f_{\mu_n}(x_n|u)\cdot g(u)\,du} \leqslant \eta^{2n}e^{-nb}\ i.o.\ \ a.s.\ \ (\textit{Algebraic Manipulation}) \tag{4.45}$$

$$\Longrightarrow \frac{\displaystyle\int_0^\rho f_{\mu_1}(x_1|u)\cdots f_{\mu_n}(x_n|u)\cdot g(u)\,du}{\displaystyle\int_0^1 f_{\mu_1}(x_1|u)\cdots f_{\mu_n}(x_n|u)\cdot g(u)\,du} \geqslant 1-\eta^{2n}e^{-nb}\ i.o.\ \ a.s.\ \ (\textit{Posterior integrates to 1}) \tag{4.46}$$

$$\begin{aligned}\Longrightarrow \eta^{2n}\frac{\displaystyle\int_\rho^1 f_{\mu_1}(x_1|u)\cdots f_{\mu_n}(x_n|u)\cdot g(u)\,du}{\displaystyle\int_0^\rho f_{\mu_1}(x_1|u)\cdots f_{\mu_n}(x_n|u)\cdot g(u)\,du} &\leqslant \frac{\eta^{4n}e^{-nb}}{1-\eta^{2n}e^{-nb}}\ \ (\textit{Using Eqns. 4.45 and 4.46})\\ &\leqslant \frac{1}{\eta^{-4n}e^{nb}-1}\ \ (\textit{Algebraic Manipulation})\end{aligned} \tag{4.47}$$

Note that $1 \leqslant \eta^4 < e^\beta$ (Using Definition 17). In the first case, we see that the Bayes Factor grows upwards towards infinity exponentially and in the second case, we see that the Bayes Factor shrinks down towards 0 exponentially. Hence, the algorithm always terminates.

Computing the Number of Samples Needed

While it is clear that the algorithm eventually terminates because of the concentration of Bayesian posteriors, it is interesting to compute the relationship between the number of samples observed by the algorithm when it terminates and the Bayes Factor T. This is particularly important for large stochastic models where each simulation involves the generation of a large number of random numbers and is expensive.

Theorem 9 *When the algorithm terminates, the upper bound on the number of samples needed to be observed is logarithmic in the Bayes Factor threshold T.*

Proof 9 *There are two cases:*

(i) Suppose the PAFM *specification is true i.e. $\rho_0 > \rho$, From the proof of the previous theorem, we know that the following holds* infinitely often almost surely,

$$\frac{\displaystyle\int_{\rho}^{1} f_{\mu_1}(x_1|u)\cdots f_{\mu_n}(x_n|u)\cdot g(u)\,du}{\displaystyle\int_{0}^{\rho} f_{\mu_1}(x_1|u)\cdots f_{\mu_n}(x_n|u)\cdot g(u)\,du} \geqslant \eta^{-2n}e^{nb}-1$$

When the algorithm terminates, we know that $\eta^{-4n}e^{nb}-1 \geqslant T$ i.e. $n \geqslant \dfrac{\log(T+1)}{b-4\log\eta}$.

(ii) Suppose the PAFM *specification is false i.e. $\rho_0 < \rho$.*

$$\frac{\displaystyle\int_{\rho}^{1} f_{\mu_1}(x_1|u)\cdots f_{\mu_n}(x_n|u)\cdot g(u)\,du}{\displaystyle\int_{0}^{\rho} f_{\mu_1}(x_1|u)\cdots f_{\mu_n}(x_n|u)\cdot g(u)\,du} \leqslant \frac{1}{\eta^{-2n}e^{nb}-1}$$

When the algorithm terminates, we know that $\dfrac{1}{\eta^{-4n}e^{nb}-1} \leqslant \dfrac{1}{T}$ i.e. $n \geqslant \dfrac{\log(T+1)}{b-4\log\eta}$.

Bounds on Type-I and Type-II Errors for Non-i.i.d. Bayesian Model Checking

While the value of Bayes Factor T is itself a sufficient stopping criteria for Bayesian Sequential Model Checking, we now relate the Bayes Factor to the frequentist notions of Type I and Type II errors. Suppose that our statistical hypothesis test stops when the lower bound on the Bayes Factor exceeds the threshold T or the upper bound on the Bayes Factor falls below the threshold $\frac{1}{T}$ (accepting the null or the alternate hypothesis respectively). We compute bounds on the Type I and Type II errors for such a test.

We now seek to characterize the Type-I/II error analytically in terms of the Bayes Factor threshold T. We assume that the actual probability of the formula being true on the model ρ_0 lies in the KL support of a proper prior and is δ-separated from the null or alternative hypothesis for some non-zero δ.

Theorem 10 *If the Bayes Factor threshold is T and the ratio of priors is γ, then the probability of the Type-I error using η-fair tests is at most $\dfrac{1}{T\ \gamma+1}$ and that of Type-II error using η-fair tests is at most $\dfrac{1}{\frac{T}{\gamma}+1}$.*

The proof is identical to the case of Bayesian statistical model checking using i.i.d. samples. We present it here for the sake of completeness. Note that Equations 4.48 and 4.49 are now justified using the definition of the Bayes Factor and Equations 4.37 and 4.39.

Proof 10 *(i) Bound on the probability of the alternate hypothesis being true:*

$$\frac{P(H_0|d)}{P(H_1|d)} = \frac{P(d|H_0)}{P(d|H_1)}\frac{P(H_0)}{P(H_1)} \qquad \ldots \textit{From Eq. 4.1}$$

$$\Longrightarrow \frac{P(H_0|d)}{P(H_1|d)} \leqslant T \ \frac{P(H_0)}{P(H_1)} \qquad \ldots \textit{Bayes Factor Threshold} \tag{4.48}$$

$$\begin{aligned}
\Longrightarrow \frac{P(H_0|d)}{P(H_1|d)} &\leqslant T \ \gamma && \ldots \textit{Since, } \gamma \textit{ is ratio of priors} \\
\Longrightarrow \frac{P(H_0|d)}{P(H_1|d)} + 1 &\leqslant T \ \gamma + 1 && \ldots \textit{Adding 1 to both sides} \\
\Longrightarrow \frac{P(H_0|d) + P(H_1|d)}{P(H_1|d)} &\leqslant T \ \gamma + 1 && \ldots \textit{Simplifying} \\
\Longrightarrow P(H_1|d) &\leqslant \frac{P(H_0|d) + P(H_1|d)}{T \ \gamma + 1} && \ldots \textit{Rearranging terms} \\
\Longrightarrow P(H_1|d) &\leqslant \frac{1}{T \ \gamma + 1} && \ldots \textit{Since, } P(H_0|d) + P(H_1|d) \leqslant 1
\end{aligned}$$

(ii) Similarly, bound on the probability of the null hypothesis being true:

$$\begin{aligned}
\frac{P(H_0|d)}{P(H_1|d)} &= \frac{P(d|H_0)}{P(d|H_1)} \frac{P(H_0)}{P(H_1)} && \ldots \textit{From Eq. 4.1} \\
\frac{P(H_0|d)}{P(H_1|d)} &\geqslant \frac{1}{T} \frac{P(H_0)}{P(H_1)} && \ldots \textit{Bayes Factor Threshold} \qquad (4.49) \\
\Longrightarrow P(H_0|d) &\leqslant \frac{1}{\frac{T}{\gamma} + 1} && \ldots \textit{Since, } P(H_1|d) + P(H_0|d) \leqslant 1
\end{aligned}$$

4.5.4 Testing Strategy for Probabilistic Reactive Modules encoding of DTMCs

We know that Girsanov's theorem [Gir60] can be used to change measures of stochastic differential equation models. In this section, we show that a similar approach can be used when dealing with DTMC models. In particular, an eventually fair testing strategy can be developed for DTMCs written as Probabilistic Reactive Modules [KNP04, BCHG$^+$97, dAKN$^+$00] which subsumes a number of modeling formalisms, such as agent based modeling.

Probabilistic Reactive Module is a precondition/postcondition based formalism for specifying probabilistic systems. When a precondiiton is true in a reactive module, the reactive module can make a transition such that the postcondition holds true after the transition. In a probabilistic reactive module, this transition is only taken with a certain probability or with a certain rate. In this section, we focus only on reactive module representations of DTMCs. In that setting, the transition is taken with a certain probability. Further, this probability distribution is assumed to be fixed in our setting. Thus, our subset of probabilistic reactive modules can be considered a factored representation of Discrete Time Markov Chains.

Example 11 (Example Probabilistic Reactive Module) *Consider a server S and a client C that talk to each other by sending messages.*

```
Module Server
Init: State = Init; MessageId=0;
State = Init: 0.95 -> State = Transmit;
State = Transmit: 0.9 -> State = Init and MessageId++;

Module Client
Init: State = Init; MessageId=0;
State = Init: 0.99 -> State = Receive;
State = Transmit: 0.99 -> State = Init and MessageId++;
```

The server has two states: Init and Transmit. It also has a variable MessageId. The server starts in the initial state and then moves on to the Transmit state with 95% probability; it stays in the Init state with 5% probability. In the transmit state, it moves to the Init state with 90

Obviously, the client is symmetric and it moves from the Init state to the Receive state with 99% probability and from the Receive state to the Init state with 99% probability.

Consider a probabilistic reactive module M with k different parameters $\lambda_1, \ldots, \lambda_k$ denoting the probability of transitions. An observable event in M is a path of length at most l in the state space of the module M.

$$s_1 \xrightarrow{p_1} s_2 \xrightarrow{p_2} \cdots \xrightarrow{p_{l-1}} s_l$$

A testing strategy can essentially perturb some of the parameters $\lambda_1, \ldots, \lambda_k$ into a new set of parameters $\lambda_1^j, \ldots, \lambda_k^j$ and then draw a random i.i.d. sample from the modified stochastic system. The probability of any given sample (observed event) such as $s_1 \xrightarrow{p_1^i} s_2 \xrightarrow{p_2^i} \cdots \xrightarrow{p_{l-1}^i} s_l$ under the probability measure μ_i is given by $p_1^i p_2^i \ldots p_{l-1}^i \equiv (\lambda_1^i)^{n_1} (\lambda_2^i)^{n_2} \ldots (\lambda_k^i)^{n_k}$. Here, n_j denotes the number of times λ_j^i occurs in the product of the transition probabilities Our testing strategy needs to be eventually fair, i.e. the geometric average $\sqrt[n]{\prod_i \left(\lambda_1^i\right)^{n_1} \left(\lambda_2^i\right)^{n_2} \ldots \left(\lambda_k^i\right)^{n_k}}$ should be bounded within an η-multiple of $(\lambda_1)^{n_1} (\lambda_2)^{n_2} \ldots (\lambda_k)^{n_k}$. One can check that the following is sufficient to guarantee the above:

$$\frac{1}{\sqrt{\eta}}\lambda_j < \sqrt[n]{\prod_{i=1}^{n} \lambda_j^i} \leqslant \sqrt{\eta}\lambda_j \qquad \text{for each } j\text{, where } 1 \leqslant j \leqslant k$$

Intuitively, this ensures that every path is equally biased against of favored in the geometric average. Now, the probability of a formula being true is just the sum of the probability of the paths satisfying the formula. Since, the inequality holds on every path individually, we can sum these inequalities to obtain the inequality for the set of paths satisfying the formula.

Thus, we have designed a sampling strategy that only needs to store the geometric average of the probability parameters and ensure that this geometric average is not too far away from the original value of the probability parameters in the reactive module description to ensure that the testing strategy is eventually fair.

4.5.5 Empirical Performance of Algorithm 3

We used our algorithm to study two simple models from computational finance: the exponential SDE model of Stock prices used in the famous Black-Scholes option pricing formula and the mean reverting Cox-Ingersoll-Ross model for interest rates.

Validation of Stock Models against High Level Properties

Consider the Stochastic Differential Equation representing the price of a stock S used in the famous Black-Scholes-Merton equation:

$$dS = m\ S\ dt \quad + \quad v\ S\ dB_t \tag{4.50}$$

where B is the Brownian motion. Here, m is the market interest rate and v is the constant volatility of the market. Intuitively, the model believes that the stock price deterministically appreciates at the same rate as the market interest rate m but has a random component called its volatility v which is a measure of the uncertainty in the price of the stock.

We use the following definition of the change of measure process Z_t as the Radon-Nikodym derivative for changing the probability measure:

$$Z_t = e^{(\alpha B_t - \frac{1}{2}\alpha^2 t)}$$

Note that α is a real constant, Z_t is a martingale, and by Girsanov's theorem [Gir60], an equivalent measure P_α is defined for every α viz., $P_\alpha = Z_t \ P$. Under the probability measure P_α, the SDE in (4.50) becomes:

$$dS = (m + \alpha v) \ S \ dt \quad + \quad v \ S \ d\hat{B}_t \tag{4.51}$$

where $\hat{B}_t$ is a Brownian Motion under the changed probability measure.

We require that the Radon-Nikodym derivatives have a geometric average in $[\frac{1}{\eta}, \eta]$, where η is small enough to permit the exponential concentration of the Bayesian posterior (See Definition 17). Starting with an initial price of \$100,000, we asked if the probability of the value of the stock being less than \$81,000 exceeded 0.00001 within a thousand time steps. The market interest rate was set at 0.05 and was held constant throughout the simulation.

$$Pr_{\geqslant 0.00001}[\mathbf{F}^{1000} \ (S < 81000)]$$

In Table 4.7, we present the number of successful samples observed by the i.i..d sampling and the Non-i.i.d. sampling algorithms respectively. The non-i.i.d. sampling exposes the rare behavior satisfying the property that the stock value falls below \$81,000. In Fig-

#Samples	#Successful (IID)	#Successful (NONIID)
10	0	0
25	0	0
50	0	0
100	0	0
500	0	0
1,000	0	1 (observed at 932^{th} sample)
2,500	0	2 (stopped at 2412^{th} sample)
5,000	0	2 (stopped at 2412^{th} sample)
7,500	0	2 (stopped at 2412^{th} sample)
10,000	1	2 (stopped at 2412^{th} sample)

Table 4.7: Number of Successful Samples observed using i.i.d. and non-i.i.d. Procedures

ure 4.19, we study the lower bound on the Bayes Factor that we derived in Theorem 9. The Bayes Factor crosses our threshold of 10^5 when we have observed about $2,400$ samples. The points where the Bayes Factor curve shows a sudden jump are the points where a confirming sample was observed by our algorithm. Our answer obtained by looking at 2,500 samples is confirmed by the i.i.d. sampling algorithm that found a confirming sample after 10,000 samples.

We also study several scenarios where the property was actually false. We expect that the number of samples needed by the non-i.i.d. sampling algorithm exceeds the number of samples needed by the i.i.d. based sampling algorithm when the rare behaviors are not needed to demonstrate the truth of our probabilistic specification. In Fig. 4.20, we plot the increase in the number of samples as a function of the maximum permissible change in the geometric mean of the change of measures η. The number of samples needed for the case of i.i.d. sampling has been normalized to 1 million samples. We also assume that the Bayes Factor is set to 10^5.

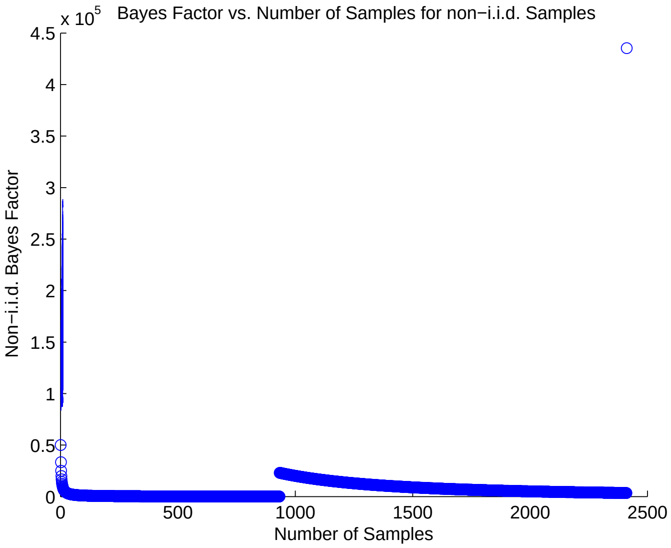


Figure 4.19: Non-i.i.d. Sampling: The property is true.

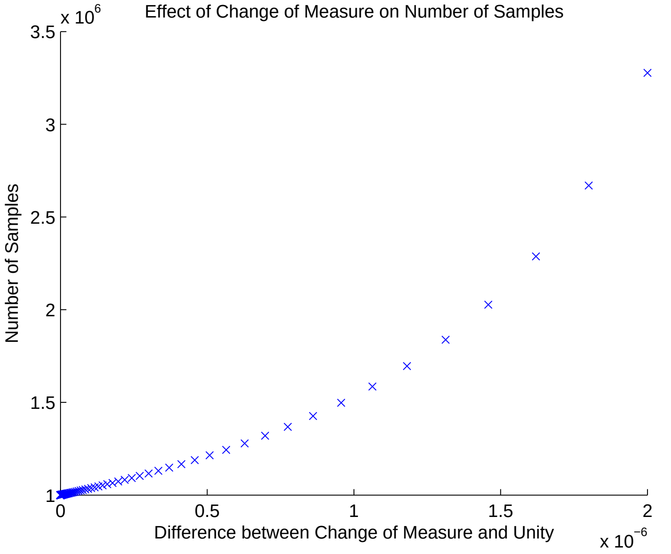


Figure 4.20: Increasing Change of Measure and Number of Samples

Validation of Cox-Ingersoll-Ross interest rate models

The Cox-Ingersoll-Ross (CIR) model is a mean reverting Stochastic Differential Equation model that captures the short term effect of market risk to the evolution of interest rates.

The model is given by the following simple SDE:

$$\frac{dr}{dt} = a(b - r)dt + v\sqrt{r}dW_t \tag{4.52}$$

The parameter a represents the speed of adjustment of the interest rate, b corresponds to the mean interest rate that the model keeps reverting to, and v represents the inherent volatility of the model and captures the notion of market risk. Starting with an initial interest rate of 0.25, we wanted to prove the property that the interest rate could fall below 10^{-8} with probability at least 10^{-4}.

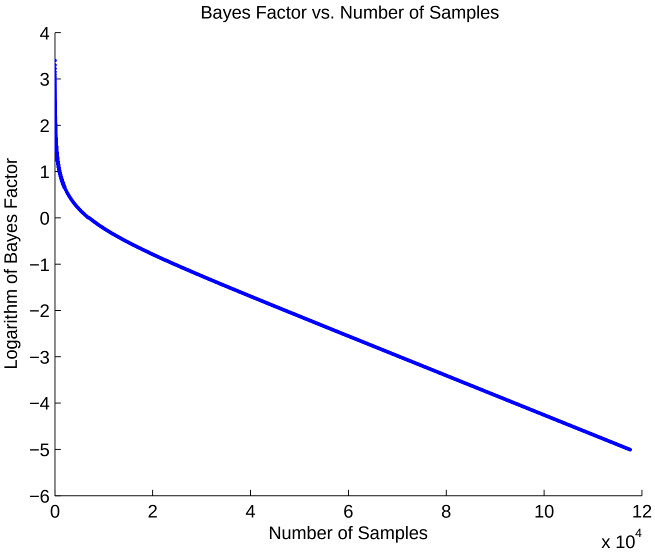


Figure 4.21: Non-i.i.d. Sampling: The property is false.

$$Pr_{\geqslant 0.0001}[\mathbf{F}^{1000}\ (r < 10^{-8})]$$

Our experiments with i.i.d. sampling algorithms indicate that this is not true. We did not see even one confirming sample in about 25,000 i.i.d. simulations of the system. We ran our non-i.i.d. simulation algorithm and the Bayes Factor plot is shown in Fig. 4.21. The non-i.i.d. algorithm also concurs that the property $\mathbf{F}^{1000}\ (r < 10^{-8})$ is not true with probability at least 0.0001.

4.6 Conclusion

We have presented three new algorithms in this chapter. The first algorithm performs Bayesian statistical model checking, and we have proved properties concerning its termination, the bounds on the number of samples needed to terminate, and frequentist bounds on errors. We then extended the algorithm to incorporate costs associated with generating samples and making a wrong prediction. Finally, we adapted our algorithm for non-i.i.d. sampling, which is important when investigating rare behaviors. We then proved properties concerning its termination, the bounds on the number of samples needed to terminate, and frequentist bounds on errors, demonstrating that the non-i.i.d. approach still provides useful guarantees.

Chapter 5

Discovery of Stochastic Biochemical Models

This chapter introduces new algorithms for discovering the kinetic parameters for *Continuous Time Markov Chain* (CTMC) models of biochemical reaction networks. Our algorithms use the fact that the probability of a measurable set of paths on such CTMCs is a uniformly continuous function of the kinetic parameters. In this chapter, we will first introduce and discuss the model discovery problem. Then, in Section 5.2, we survey the class of CTMC models that we are interested in. We present a Survey Sampling based approach to Statistical Model Checking in Section 5.3. We will then present results that characterize the probability with which an AFM specification is true on a model in Section 5.4. Finally, we discuss model discovery algortihms in Section 5.5 and show results on representative benchmarks in Section 5.6.

5.1 Introduction

A biochemical reaction network consists of a number of distinct molecular species that interact dynamically according to a prescribed set of reaction rules. Each rule describes the production, consumption, or transformation of a subset of species. For example, the rule

$$A + B \xrightarrow{k} A' + B$$

describes the transformation of species A into A'. The transformation is mediated by B, but B is otherwise unaffected by the process. The transformation occurs with a forward reaction rate k.

Ordinary differential equations are often used to model biochemical systems. These equation-based models are deterministic approximations to what is inherently a stochastic process. In contrast, *Continuous Time Markov Chain* (CTMC) models precisely represent the stochastic nature of biochemical interactions [Gil77, Gil07, RPCG03]. Here, the state transitions in the CTMC model correspond to discrete changes in the number of copies of each species, due to the execution of a reaction rule. The rates of the stochastic state transitions are derived from the kinetic reaction rate constants in the biochemical reactions. One of the most difficult challenges that arises in the development of reaction network models is the discovery of kinetic reaction rate parameters that are consistent with the empirical behavior of the biochemical system being modeled. In the biological modeling literature, there are two common strategies for obtaining model parameters — *measurement* and *estimation*. Rate parameters can be measured experimentally using, for example, calorimetry or spectroscopy. Unfortunately, it can be difficult or even impossible to measure *all* the

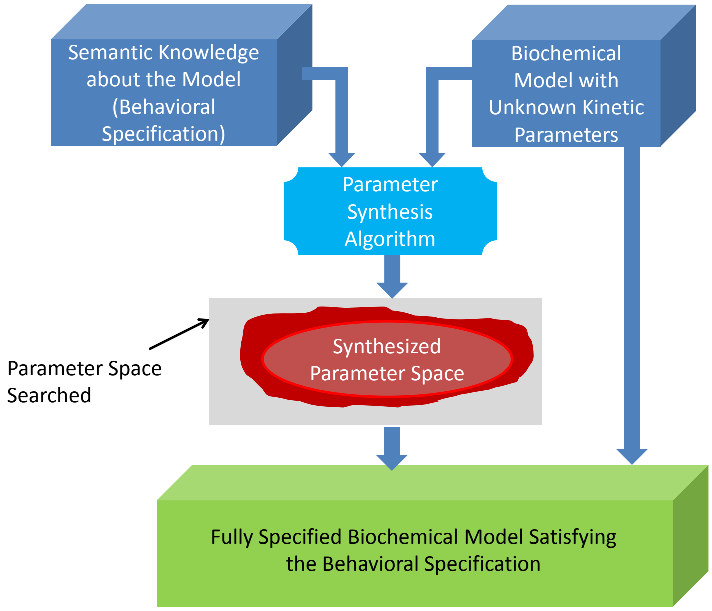

Semantic Knowledge about the Model (Behavioral Specification)
Biochemical Model with Unknown Kinetic Parameters
Parameter Synthesis Algorithm
Parameter Space Searched
Synthesized Parameter Space
Fully Specified Biochemical Model Satisfying the Behavioral Specification

Figure 5.1: Parameter Synthesis Problem

rate constants specified in a given model, especially those between short-lived, transient species. Thus, parameter estimation is generally used to calibrate the model. This process involves a time-consuming and often an *ad hoc* manual search through the parameter space. The goal of the search is to calibrate the model to one or more empirically observed *high-level behaviors* from the real system (e.g., turnover rates).

In this chapter, we consider an alternative strategy to manual model calibration — *automated parameter synthesis* (Fig. 5.1). Biologists frequently summarize experimental data with high-level behavioral descriptions (e.g., “the system is bi-stable”). Modelers generally want their models to satisfy these high-level specifications and not the precise details of any particular set of experimental observations (due to stochastic dynamics, measurement errors, etc.). Parameter synthesis is the task of identifying the volume in parameter space that gives rise to the prescribed behavior. Synthesis is a more challenging problem than parameter estimation, because every possible parameter combination must be characterized as either satisfying the property, or not. Synthesizing parameters for CTMCs is especially challenging because of the stochastic nature of the model, and because there are an *uncountably infinite* number of parameter combinations.

We introduce three algorithms for studying the parameter space of stochastic biochemical models. The first two algorithms solve the synthesis problem for CTMCs using a combination of statistical model validation (Chapter 4) and *abstraction refinement*. Specifically, given a CTMC, $\mathcal{M}$, with unknown parameters, a property specification, ϕ, and a probability threshold, ρ, our algorithms obtain a region in parameter space, $\tilde{\mathcal{V}}_{\phi,\rho}$, such that for any choice of parameters $\theta \in \tilde{\mathcal{V}}_{\phi,\rho}$ the resulting model, $\mathcal{M}_\theta$, will satisfy ϕ with probability at least ρ. That is, $\mathcal{M}_\theta \models Pr_{\geqslant\rho}(\phi)$. Additionally, for any choice of parameters such that $\theta \notin \tilde{\mathcal{V}}_{\phi,\rho}$, $\mathcal{M}_\theta \not\models Pr_{\geqslant\frac{\rho}{\eta}}(\phi)$ where $\eta > 1$ is an input parameter to the synthesis algorithm that controls the quality of the synthesized parameter space. The computed volume is an approximation to the true volume satisfying the specification, $\mathcal{V}$, but the difference between $\tilde{\mathcal{V}}$ and $\mathcal{V}$ can be made arbitrarily small at the expense of additional computational resources (by making η closer to one). The second algorithm is a modified version of the first, and uses *monotonicity* and *abstraction refinement* to accelerate the synthesis. The third algorithm performs a parameter *search* and finds an approximation to the single parameter combination that maximizes the probability that the property holds. All three algorithms can also decide whether a given probabilistic property is *infeasible*. That is, there is no choice of parameters that satisfy the given behavioral specification with probability at least ρ.

The specific contributions of this chapter are as follows:

1. We present an automated algorithm for the calibration of stochastic biochemical models against high-level behavioral specifications of the underlying biochemical system. The proofs underlying our algorithm use a new statistical model validation algorithm based on survey sampling. To the best of our knowledge, our use of survey sampling based statistical model validation is a novel method of constructing correctness proofs for synthesis algorithms of stochastic biochemical models. Our survey sampling based approach to statistical model validation is the cornerstone of our proofs as it allows us to reason about sample paths from all the possible parameterizations of a given model, without worrying about changes in the probability measures of the individual paths.
2. We show that the logarithm of the probability of an adapted finitely monitorable formula (see Ch. 2, Sec. 2.2) being true on a model is *uniformly and jointly continuous* with respect to the kinetic parameters in the stochastic biochemical model. The correctness results of our algorithms use this property.
3. We apply our algorithm to synthesize up to 6 parameters simultaneously for a Fibroblast Growth Factor signaling model studied in literature [HKN$^+$06, HKN$^+$08]. We note that this is the first time that the synthesis of as many as six parameters for stochastic models against probabilistic temporal logic properties has been reported. We also demonstrate our ability to show infeasibility of probabilistic adapted finitely monitorable specifications on a biochemical model in a given parameter range. Finally, when parameter estimation is desired, as opposed to synthesis, we show that our approach can scale to even higher dimensional spaces, by identifying the single

parameter combination that maximizes the probability of the formula being true in an 11-dimensional system.

5.2 Stochastic Models of Biochemical Systems

The stochastic process underlying a dynamic biochemical system can be modeled with a Continuous Time Markov Chain (CTMC) [Gil77, Gil07, RPCG03]. The key arguments behind the underlying theoretical derivation are that the state of a chemical system changes only through the completion of a chemical reaction, and that the probability of a chemical reaction occurring during a given interval of time is proportional to the probability that the reactants come together with enough energy to overcome the reaction's activation energy barrier. We formally define the model on which our algorithms operate.

Our algorithms can be applied to CTMCs of a particular form. Namely, those modeling the dynamics of biochemical reactions. We consider a biochemical reaction system with n biochemical reactions $r_1, r_2, \ldots, r_n$ among m biochemical species $X_1, X_2, \ldots, X_m$:

$$\begin{array}{rccc}
r_1: & \alpha_1^1 X_1 + \cdots + \alpha_m^1 X_m & \xrightarrow{k_1} & \alpha'^1_1 X_1 + \cdots + \alpha'^1_m X_m \\
r_2: & \alpha_1^2 X_1 + \cdots + \alpha_m^2 X_m & \xrightarrow{k_2} & \alpha'^2_1 X_1 + \cdots + \alpha'^2_m X_m \\
 & \ldots \quad \ldots \quad \ldots & \xrightarrow{k_j} & \ldots \quad \ldots \quad \ldots \\
r_n: & \alpha_1^n X_1 + \cdots + \alpha_m^n X_m & \xrightarrow{k_n} & \alpha'^n_1 X_1 + \cdots + \alpha'^n_m X_m
\end{array}$$

The stoichiometric constants $\alpha_1^1, \ldots, \alpha_m^n, \alpha'^1_1, \ldots, \alpha'^n_m$ are non-negative integers, and $k_i \in$

$\mathbb{R}^+$ $(1 \leqslant i \leqslant n)$ are reaction rates. The labeled Continuous Time Markov Chain corresponding to the biochemical system is the three tuple (S, R, L) where:

- $S = \{0, 1, 2, \ldots N\} \times \{0, 1, 2, \ldots N\} \times \cdots \times \{0, 1, 2, \ldots N\}$ (m times). Here, N denotes the maximum number of copies of any biochemical species as the system evolves with time.
- $L(s) = s \equiv (x_1(s), \ldots, x_m(s))$ where $x_j(s) \in [0, N]$ denotes the number of entities of biochemical species X_j.
- $R(s, s') = \begin{cases} k_j x_1(s)^{\alpha_1^j} \ldots x_m(s)^{\alpha_m^j} & \textit{if } s' \textit{ results from executing reaction } r_j \textit{ in state } s \\ 0 & \textit{otherwise} \end{cases}$

The probability density of a transition from state s to state s' after spending time Δ in s, denoted by $P(s, s', \Delta)$, is $k_j x_1(s)^{\alpha_1^j} \ldots x_m(s)^{\alpha_m^j}$ $(e^{-\sum_{i=1}^{n} \mathcal{I}(r_i) k_i x_1(s)^{\alpha_1^i} \ldots x_m(s)^{\alpha_m^i} \Delta_i})$. The indicator function $\mathcal{I}(r_i)$ is 1 if and only if it is possible to execute the reaction r_i in the state s; otherwise, the indicator function $\mathcal{I}(r_i)$ is zero.

Each state in this discrete state-space model for biochemical systems is labeled with the number of copies of various biochemical species in the biochemical system. No two distinct states have exactly the same number of copies of various biochemical species. The rate of transition from one state s to another state s' is proportional to the number of copies of each biochemical species raised to the stoichiometric coefficient for that species in the reaction r_j that takes the system from state s to state s'. The constant of proportionality is given by the rate constant k_j. Note that there is at most one biochemical reaction that can take the system from a given state to another given state in a single transition.

5.3 Statistical Model Validation for Model Disovery

Our parameter synthesis algorithms use statistical model validation algorithms to solve key subproblems. Additionally, we will be using statistical approaches to develop our proofs. Instead of proving properties of an arbitrary (possibly infinite) set of paths, we will use the notion of *unbiased statistical estimators* to develop the correctness arguments of our results. In particular, we invoke *survey sampling*. Our survey sampling based approach to statistical model validation is the cornerstone of our proofs as it allows us to reason about sample paths from all the possible parameterizations of a given model, without worrying about changes in the probability measures of the individual paths.

We will show in Sec. 5.4 that survey sampling leads to a powerful new proof technique for stochastic systems where one analytically lifts proof arguments from one parameterization of a stochastic system to another parameterization of the same stochastic system. The two stochastic systems are identical, except for a change of probability measure on the paths. We believe this new proof technique is perhaps one of the most interesting contributions of this chapter.

In the rest of the chapter, we are only interested in knowing whether a stochastic model satisfies a probabilistic adapted finitely monitorable logic formula with arbitrary high confidence. That is, deciding whether $\mathcal{M} \models_q P_{\geqslant\rho}(\phi)$ or $\mathcal{M} \models_q P_{<\rho}(\phi)$ is true. Here, q is the *confidence probability*; it represents our confidence that our decision is correct. Statistical model validation algorithms generally terminate with a decision when the probability of Type-I/Type-II errors have fallen below a threshold such that the user specified confidence probability q is respected.

There are a variety of statistical model validation algorithms from which to choose, and our parameter synthesis algorithm itself is agnostic to the particular choice of the statistical model validation algorithm. In Chapters 3 and 4, we discussed a number of existing statistical model validation algorithms. We now introduce an additional statistical model validation algorithm that is important for the construction of our proofs in this chapter.

5.3.1 Survey Sampling based Statistical Model Validation

The common feature of the model validation algorithms in Chapters 3 and 4 is that they sample from the set of unique trajectories defined by a stochastic model with known parameters (or prior distributions over parameters, for the Bayesian methods). These samples are then used to reject the null or the alternate hypothesis or estimate the actual probability with which a specification holds on the stochastic model. When the parameters (or priors over parameters) of the model are given, then the appropriate sampling strategy is to generate sample trajectories according to the underlying probability distribution implied by the parameters. That is, trajectories with higher probability tend to be sampled with higher frequency than those with lower probability.

Our goal in this chapter is different. We seek to synthesize parameters for a given incompletely specified stochastic model. Under these circumstances, we argue that *survey sampling* [SO96] based on simple random sampling with replacement is the appropriate theoretical framework for generating independent and identically distributed (i.i.d.) trajectories from a parameterized family of models. In contrast to the sampling strategies employed in other statistical model checking techniques, survey sampling draws samples *uniformly* from all unique traces of the model. That is, the samples are generated without respect to any particular choice of kinetic parameters. *The fact that the samples are not affected by the probability space on the paths is crucial to the construction of our proofs and correctness arguments.* We note that survey sampling [SO96] is primarily a mechanism to construct the proofs; the actual computation during the synthesis may use any of the aforementioned statistical model validation algorithms.

Suppose that n samples have been uniformly drawn from the unique traces of the model. Then, for any choice of parameters, θ, we can (retroactively) label each trace σ_i of the model $\mathcal{M}(\theta)$ with a probability density value $P_\theta(\sigma_i)$. Each trace is also labeled with an indicator variable $\mathcal{I}(\sigma_i)$ such that the indicator is 1 if the trace satisfies the given adapted finitely monitorable property, and 0 otherwise. Given these, we define the random variable $X_i \equiv P_\theta(\sigma_i)\mathcal{I}(\sigma_i)$. Thus, the sample mean or expected value of X_i (taken over the n samples) is an unbiased estimator of the probability that the model will satisfy the adapted finitely monitorable formula. We recall that the bias of an estimator is the difference between that estimator's expected value, and the true value of the parameter being estimated. A zero bias estimator is said to be unbiased.

$$\overline{X} = E[X] = \frac{1}{n}\sum_{i=1}^{n} X_i$$

The variance of the sample mean, $Var(\overline{X})$, can be computed as follows:

$$Var(\overline{X}) = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} Cov(X_i, X_j) = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} Var(X_i) = \frac{\sigma^2}{n}$$

The problem with this computation is that it expresses the variance of the mean of observed sample traces in terms of the variance of the *entire* population of unique traces in the system (σ) - the latter quantity is not readily computable. Hence, we would like to express the variance of the sample traces in terms of computable statistics of the observed traces themselves. An unbiased estimator of $Var(\overline{X})$ is given by:

$$s_{\overline{X}}^2 = \frac{\hat{\sigma}^2}{N}\left(\frac{N-n}{n-1}\right)$$

where $\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(X_i - \overline{X})^2$ and N is the total number of samples.

We have shown how the mean and the standard deviation of the sampling distribution of $\overline{X}$ can be computed. The central limit theorem can now be used to show that the sampling distribution can be approximated using a Gaussian distribution. By the central limit theorem,

$$P\left(\frac{\overline{X}_n - \mu}{\sigma\sqrt{n}} \leqslant z\right) \rightarrow \Phi(z) \qquad \text{as } n \rightarrow \infty$$

where Φ is the cumulative distribution function of the standard normal distribution. To summarize, if we have drawn n samples (and n is large) out of the N possible samples, the distribution of the sample traces is given by the normal distribution:

$$\mathcal{N}(\overline{X}, \frac{N-n}{n(n-1)N}\sum_{i=1}^{n}(X_i - \overline{X})^2)$$

Thus, survey sampling estimates the mean and the variance of the probability of the property being true for the model. We note that the exact form of these expressions or the computation of the sample mean and variance of the sample mean is not a subject of key interest in this chapter. *The fact that survey sampling can be used for statistical model validation and the samples involved in survey sampling do not change with change in probabiliity measures of a model is used to argue the correctness of our algorithms.*

5.4 Problem Statement and Theorems

Given a parameterized biochemical system $\mathcal{M}(\Theta)$ with unknown kinetic parameters Θ and a probabilistic adapted finitely monitorable logic formula $Pr_{\geqslant\rho}(\phi)$, the parameter synthesis problem is to discover a bounded region in parameter space, $\mathcal{V}$, such that the system $\mathcal{M}$ with parameter values $\theta \equiv (k_1, \ldots, k_n) \in \mathcal{V}$, i.e. $\mathcal{M}(\theta)$, satisfies the formula.

A brute-force approach to solving the synthesis problem would involve exhaustively searching the space of all parameter values and using statistical model validation to estimate the probability that each parameter combination results in a model that satisfies the formula. Unfortunately, a brute-force algorithm will not terminate because the search space for the parameter values is uncountably infinite. An alternative approach is to discretize the parameter space and sample from the resulting finite search space. However, two questions remain open:

I. Can we bound the probability of the formula ϕ being true on the model $\mathcal{M}$ in a *dense set* of parameters by sampling *only finite points* in this dense set?

II. What is a *good* discretization of the space of parameters?

This chapter provides an affirmative answer to the first question. Then, we present a methodology to address the second problem by developing a new theoretical characterization for the probability of a formula being true as a function of the reaction rate parameters. We will show that bounded changes in the logarithm of reaction rates make bounded changes in the logarithm of the probability density associated with any *finite path* of non-zero probability measure, and that this change can be made arbitrary small by choosing a sufficiently small change in the reaction rate parameters. To do this, we will use the *uniform continuity* of the logarithm of the probability density of a path in a stochastic biochemical model with respect to the logarithm of the reaction rate parameters in a bounded parameter space to prove the correctness of our synthesis algorithms.

5.4.1 Uniform Continuity in the Logarithmic Parameter Space

The change in the logarithm of the probability density associated with any finite path of a biochemical stochastic model can be bounded by a function of the change in the logarithm of the reaction rate (kinetic) parameters. Moreover, this change in the logarithm of the probability density can be made arbitrarily small by choosing a sufficiently small change in the logarithm of the reaction rate parameters.

Theorem 11 *If* $k_j, k'_j \in (0, M]$ *and* $|\log k'_j - \log k_j| \leqslant \delta$, $|k'_j - k_j| \leqslant M(e^\delta - 1)$.

Proof 11 *Without loss of generality, assume* $k_j > k'_j$.

$$\begin{aligned}
&|\log k'_j - \log k_j| && \leqslant \delta \\
\Rightarrow &\log k_j - \log k'_j && \leqslant \delta \\
\Rightarrow &\log \frac{k_j}{k'_j} && \leqslant \delta \\
\Rightarrow &\frac{k_j}{k'_j} && \leqslant e^\delta
\end{aligned}$$

$$\Rightarrow k_j \leqslant k'_j\ e^{\delta} \quad (\textit{Taking exponential on both sides})$$

$$\Rightarrow k_j - k'_j \leqslant k'_j\ \left(e^{\delta} - 1\right) \quad (\textit{Subtracting } k'_j \textit{ from both sides})$$

$$\Rightarrow |k_j - k'_j| \leqslant k'_j\ \left(e^{\delta} - 1\right)$$

$$\Rightarrow |k_j - k'_j| \leqslant M\ \left(e^{\delta} - 1\right) \quad (\textit{Since } k'_j \leqslant M)$$

Consider a path σ in a biochemical stochastic system $\mathcal{M}(\Theta)$. We recall that Θ denotes the reaction rate parameters. Further, let $P(\sigma)$ denote the probability density associated with the path in $\mathcal{M}(\theta)$ while $P'(\sigma)$ denotes the probability density of the path in $\mathcal{M}(\theta')$. We now show that the difference between the logarithm of $P(\sigma)$ and the logarithm of $P'(\sigma)$ can be made as small as needed by making the difference between θ and θ' sufficiently small.

Theorem 12 (Uniform Continuity of Path Probability Density in Parameter Space) *For every $\epsilon \in \mathcal{R}^+$, there exists $\delta \in \mathcal{R}^+$ such that $|\log P'(\sigma) - \log P(\sigma)| \leqslant \epsilon$ holds whenever $|\log k'_j - \log k_j| \leqslant \delta$, for all j $(1 \leqslant j \leqslant n)$.*

Proof 12 *We present a detailed proof in Appendix section A.2. Here, we present an intuitive sketch of the proof. We know that the probability density of moving from state s_i to state s_{i+1} by executing reaction r_{j_i} after time Δ_i is given by*

$$P(s_i \xrightarrow{\Delta_i} s_{i+1}) = k_{j_i} x_1(s_i)^{\alpha_1^{j_i}} \ldots x_m(s_i)^{\alpha_m^{j_i}} \text{exp} \left(- \sum_{h=1}^{n} \mathcal{I}(r_h, i) k_h x_1(s_i)^{\alpha_1^h} \ldots x_m(s_i)^{\alpha_m^h} \Delta_i \right)$$

Taking logarithms on both sides,

$$\log P(s_i \xrightarrow{\Delta_i} s_{i+1})$$

$$\begin{aligned} &= \log\left(k_{j_i} x_1(s_i)^{\alpha_1^{j_i}} \dots x_m(s_i)^{\alpha_m^{j_i}}\right) - \sum_{h=1}^{n} \mathcal{I}(r_h, i) k_h x_1(s_i)^{\alpha_1^h} \dots x_m(s_i)^{\alpha_m^h} \Delta_i \\ &= \log\left(k_{j_i}\gamma^{j_i}_{(i,i+1)}\right) - \sum_{h=1}^{n} k_h \mathcal{I}(r_h,i)\gamma^h_{(i,i+1)}\Delta_i \end{aligned}$$

Here, $\gamma^h_{(i,i+1)} \overset{def}{\equiv} x_1(s_i)^{\alpha_1^h}\dots x_m(s_i)^{\alpha_m^h}$ *is a quantity independent of* k_h $(1 \leqslant h \leqslant n)$*. And so,* $|\log P(s_i \xrightarrow{\Delta_i} s_{i+1}) - \log P'(s_i \xrightarrow{\Delta_i} s_{i+1})|$

$$\begin{aligned} &= \ \left| \ \log k_{j_i} - \log k'_{j_i} + \sum_{h=1}^{n} \mathcal{I}(r_h,i)(k'_h - k_h)\gamma^h_{(i,i+1)}\Delta_i \ \right| \\ &\leqslant \quad |\log k_{j_i} - \log k'_{j_i}| + \gamma^{max}_{(i,i+1)}\Delta_i \sum_{h=1}^{n} |k'_h - k_h| \ (\gamma^{max}_{(i,i+1)} \overset{def}{\equiv} \max_{1 \leqslant h \leqslant n} \mathcal{I}(r_h,i)\gamma^h_{(i,i+1)}, \ \textit{Triangle Ineq.}) \end{aligned}$$

Consider the finite path $\sigma \equiv s_0 \xrightarrow{\Delta_0} \cdots \xrightarrow{\Delta_{l-1}} s_l$*. Let* $P(\sigma)$ *be the probability density associated with the path in the model* $\mathcal{M}(\theta)$ *and* $P'(\sigma)$ *be the probability density associated with the model* $\mathcal{M}(\theta')$*. We know that* $P(\sigma) = P(s_0 \xrightarrow{\Delta_0} s_1) \times \cdots \times P(s_{l-1} \xrightarrow{\Delta_{l-1}} s_l)$*.*

So, $| \ \log P(\sigma) - \log P'(\sigma) \ |$

$$\begin{aligned} &\leqslant \quad \sum_{i=0}^{l-1} |\log k_{j_i} - \log k'_{j_i}| + \sum_{i=0}^{l-1}\left(\gamma^{max}_{(i,i+1)}\Delta_i \sum_{h=1}^{n} |k'_h - k_h|\right) \qquad \dots \textit{Triangle Inequality} \\ &\leqslant \quad l \max_{j_i, i \in [0,l-1]} |\log k_{j_i} - \log k'_{j_i}| + \left(\sum_{h=1}^{n} |k'_h - k_h|\right) \sum_{i=0}^{l-1}\left(\gamma^{max}_{(i,i+1)}\Delta_i\right) \qquad \dots \textit{Algebraic Manipulation} \\ &\leqslant \quad l \ |\log k_j - \log k'_j|^{max} + \gamma^{max}\left(\sum_{h=1}^{n} |k'_h - k_h|\right) \sum_{i=0}^{l-1} \Delta_i \end{aligned}$$

where $\gamma^{max} \overset{def}{\equiv} \max_{i \in [0,l-1]} \gamma^{max}_{(i,i+1)}$ *and* $|\log k_j - \log k'_j|^{max} \overset{def}{\equiv} \max_{j_i, i \in [0,l-1]} |\log k_{j_i} - \log k'_{j_i}|$*. And* *so,* $| \ \log P(\sigma) - \log P'(\sigma) \ |$

$$\begin{aligned}
&\leqslant \quad l|\log k_j - \log k'_j|^{max} + \gamma^{max}\Big(\sum_{h=1}^{n}|k'_h - k_h|\Big)\Delta_{total} && \ldots \Delta_{total} \equiv \sum_{i=0}^{l-1}\Delta_i \\
&\leqslant \quad l \ \ \delta + \gamma^{max}\Big(\sum_{h=1}^{n} M\left(e^{\delta}-1\right)\Big)\Delta_{total} && \ldots \textit{From Lemma 11}
\end{aligned}$$

To show that $| \ \log P(\sigma) - \log P'(\sigma) \ | \leqslant \epsilon$*, it is sufficient to show that the following holds:*

$$l \ \ \delta + \gamma^{max}\Big(\sum_{h=1}^{n} M\left(e^{\delta}-1\right)\Big)\Delta_{total} \leqslant \epsilon$$

From the statement of our theorem, we know that $|\log k_j - \log k'_j|^{max} \leqslant \delta$*. One can verify that the following choice of* δ *is sufficient to show that* $|\log P(\sigma) - \log P'(\sigma)| \leqslant \epsilon$*:*

$$\begin{aligned}
\delta &= \min\left(\frac{\epsilon}{l(n+1)}, \log\left(\frac{\epsilon}{(n+1)\max(\gamma^{max} M \Delta_{total}, 1)} + 1\right)\right) \\
&\overset{def}{\equiv} \delta(\epsilon, \mathcal{M}).
\end{aligned}$$

In the rest of the chapter, we will use the notation $\delta(\epsilon, \mathcal{M})$ *to denote this value of* δ*.*

The uniform continuity arguments we have presented allow us to establish results on a *finite* set of points in a bounded parameter space and then extend the statements of these results to the entire *uncountably infinite* parameter space. A natural follow-up investigation is to characterize the probability of a formula being true as a function on the parameter space. In the following lemma, we define an unbiased statistical estimator of the probability of a formula being true on a model.

Lemma 1 *[Survey Sampling based Unbiased Statistical Estimator for the Probability of a Finite Set of Paths] Given a finite set of independent and identically distributed (i.i.d.) sample paths $\sigma_1, \sigma_2, \ldots, \sigma_T$ (of length at most l) drawn uniformly from a (possibly infinite) set of paths $\mathcal{T}$ such that each path is labeled with either 0 or 1 i.e. $L(\sigma_i) \in \{0, 1\}$, an unbiased statistical estimator for the probability of the set of paths with label 1 in $\mathcal{T}$ is given by*

$$\hat{P} \overset{def}{\equiv} \frac{\sum_{L(\sigma_t)=1, 1 \leqslant t \leqslant T} P(\sigma_t)}{\sum_{t=1}^{T} P(\sigma_t)}$$

i.e.

$$\hat{P} \overset{def}{\equiv} \frac{\sum_{L(\sigma_t)=1, 1 \leqslant t \leqslant T} \prod_{i=1}^{l} k_{j_i}(\sigma_t) \gamma^{j_i}_{(i,i+1)}(\sigma_t) \exp(-\sum_{h=1}^{n} \mathcal{I}(r_h, i, \sigma_t) k_h(\sigma_t) \gamma^{h}_{(i,i+1)}(\sigma_t) \Delta_i(\sigma_t))}{\sum_{t=1}^{T} \prod_{i=1}^{l} \mathcal{I}(r_h, i, \sigma_t) k_{j_i}(\sigma_t) \gamma^{j_i}_{(i,i+1)}(\sigma_t) \exp(-\sum_{h=1}^{n} k_h(\sigma_t) \gamma^{h}_{(i,i+1)}(\sigma_t) \Delta_i(\sigma_t))}$$

Here, $k_{j_i}(\sigma_t)$, $\gamma^{j_i}_{(i,i+1)}(\sigma_t)$, and $\Delta_i(\sigma_t)$ represent the values of k_{j_i}, $\gamma^{j_i}_{(i,i+1)}$ and Δ_i corresponding to the path σ_t. Also, $\gamma^{h}_{(i,i+1)}(\sigma_t) \overset{def}{\equiv} x_1(s_i(\sigma_t))^{\alpha^h_1} \ldots x_m(s_i(\sigma_t))^{\alpha^h_m}$ is a quantity independent of $k_h(\sigma_t)$ $(0 \leqslant h \leqslant n, 1 \leqslant t \leqslant T)$. The indicator function $\mathcal{I}(r_h, i, \sigma_t)$ indicated whether the reaction r_h was fired at the i^{th} step in the path σ_t

Theorem 13 (Uniform Continuity of the Unbiased Estimator) *Let $\hat{P}$ be the unbiased statistical estimator of the probability with which an AFM specification ϕ is true on the model $\mathcal{M}(\theta)$ and $\hat{P}'$ be the unbiased statistical estimator of the probability with which ϕ is true on the model $\mathcal{M}(\theta')$. For every $\epsilon \in \mathcal{R}^+$, there exists $\delta \in \mathcal{R}^+$ such that $|\log \hat{P}' - \log \hat{P}| \leqslant \epsilon$ holds whenever $|\log k'_j - \log k_j| \leqslant \delta$, for all j $(1 \leqslant j \leqslant n)$.*

Proof 13 *Consider a survey sampling based unbiased statistical estimator of the probabil-*

ity of the formula using T samples $\sigma_1, \sigma_2 \ldots \sigma_T$. For any $\epsilon/2 \in \mathbb{R}^+$, we know that there exists $\delta_i \in \mathbb{R}^+$ such that $|\log P'(\sigma_i) - \log P(\sigma_i)| \leqslant \frac{\epsilon}{2}$ holds whenever $|\log k_j{}' - \log k_j| \leqslant \delta_i$. Choose δ as the smallest of all δ_i ($1 \leqslant i \leqslant T$).

$$\begin{aligned}
\hat{P}' = & \quad \frac{\sum_{\sigma \models \phi} P'(\sigma)}{\sum_{\sigma} P'(\sigma)} && \textit{Statistical Estimator Definition} \\
\leqslant & \quad \frac{\sum_{\sigma \models \phi} e^{\frac{\epsilon}{2}} P(\sigma)}{\sum_{\sigma} e^{-\frac{\epsilon}{2}} P(\sigma)} && \textit{Uniform Continuity of Paths} \\
= & \quad e^{\epsilon} \frac{\sum_{\sigma \models \phi} P(\sigma')}{\sum_{\sigma} P(\sigma')} && \textit{Algebraic Manipulation} \\
= & \quad e^{\epsilon} \hat{P} && \textit{Statistical Estimator Definition} \\
\implies \log \hat{P}' - \log \hat{P} \leqslant \epsilon & && \textit{Taking log on both sides}
\end{aligned}$$

Similarly, we can also argue that

$$\begin{aligned}
\hat{P}' = & \quad \frac{\sum_{\sigma \models \phi} P'(\sigma)}{\sum_{\sigma} P'(\sigma)} && \textit{Statistical Estimator Definition} \\
\geqslant & \quad \frac{\sum_{\sigma \models \phi} e^{\frac{-\epsilon}{2}} P(\sigma)}{\sum_{\sigma} e^{\frac{\epsilon}{2}} P(\sigma)} && \textit{Uniform Continuity of Paths} \\
= & \quad e^{-\epsilon} \frac{\sum_{\sigma \models \phi} P(\sigma')}{\sum_{\sigma} P(\sigma')} && \textit{Algebraic Manipulation} \\
= & \quad e^{-\epsilon} \hat{P} && \textit{Statistical Estimator Definition} \\
\implies \log \hat{P}' - \log \hat{P} \geqslant -\epsilon & && \textit{Taking log on both sides}
\end{aligned}$$

Thus, we know that $|\log \hat{P}' - \log \hat{P}| \leqslant \epsilon$. Hence, the logarithm of the unbiased statistical estimator of the probability of the formula is uniformly continuous in the kinetic parameter space.

If one could show that the probability of a formula being true is monotonic as a function on the parameter space, it would be possible to develop abstraction refinement algorithms [CGJ$^+$00, JBS07] by sampling with varying discretizations of the parameter space. In the following theorem, we show that the probability density of a path is *not* necessarily monotonic in the parameter space and compute the point in the parameter space where the extremum of the probability density of a path is reached.

Theorem 14 (Non-Monotonicity of Path Probability Density in the Parameter Space) *The probability density of a path in a stochastic biochemical model is* not *necessarily monotonic in the parameter space.*

Proof 14

$$\begin{aligned}
P(\sigma) &= P(s_0 \xrightarrow{\Delta_0} s_1) \times \cdots \times P(s_{l-1} \xrightarrow{\Delta_{l-1}} s_l) \\
\Rightarrow \log P(\sigma) &= \log P(s_0 \xrightarrow{\Delta_0} s_1) + \cdots + \log P(s_{l-1} \xrightarrow{\Delta_{l-1}} s_l) \\
\Rightarrow \frac{1}{P(\sigma)} \frac{dP(\sigma)}{dk_u} &= \frac{d}{dk_u} \sum_{i=0}^{l-1} \big(\log k_{j_i} + \log \big(\gamma^{j_i}_{(i,i+1)}\big) - \sum_{h=1}^{n} \mathcal{I}(r_h, i) k_h \gamma^{h}_{(i,i+1)} \Delta_i \big) \\
& \qquad\qquad (r_{j_i} \textit{ is reaction at step i.}) \\
\Rightarrow \frac{dP(\sigma)}{dk_u} &= P(\sigma) \sum_{i=0}^{l-1} \left(\mathcal{I}(u = j_i) \frac{1}{k_u} - \mathcal{I}(r_u, i) \gamma^{u}_{(i,i+1)} \Delta_i \right) \\
& \qquad\qquad (\mathcal{I} \textit{ is the indicator function.})
\end{aligned}$$

Clearly, $\frac{dP(\sigma)}{dk_u}$ can be either positive or negative depending upon the path in consideration and the value of the kinetic parameters. Continuity ensures that the function $\frac{dP(\sigma)}{dk_u}$ is zero *for some value of k_u.*

Theorem 15 (Extremum of the Probability Density of a Path) *The probability density of a path $\sigma \equiv s_0 \xrightarrow{\Delta_0} s_1 \xrightarrow{\Delta_1} s_2 \cdots \xrightarrow{\Delta_{l-1}} s_l$ in the stochastic model attains a unique extremum at the point $(k_1^{extrema}, \ldots, k_n^{extrema})$, where*

$$k_u^{extrema} = \frac{N^{k_u}}{\sum_{i=0}^{l-1} \mathcal{I}(r_u, i)\gamma_{(i,i+1)}^{u}\Delta_i}$$

and N^{k_u} is the number of times the reaction r_u is executed along the path σ.

Proof 15

$$\begin{aligned} \log P(\sigma) &= \log P(s_0 \xrightarrow{\Delta_0} s_1) + \cdots + \log P(s_{l-1} \xrightarrow{\Delta_{l-1}} s_l) \\ &= \sum_{i=1}^{l-1} \log k_{j_i} + \sum_{i=1}^{l-1} \log\left(\gamma_{(i,i+1)}^{j_i}\right) - \sum_{i=1}^{l-1}\sum_{h=1}^{n} \mathcal{I}(r_h, i) k_h \gamma_{(i,i+1)}^{h} \Delta_i \end{aligned}$$

Partially differentiating with respect to each reaction rate parameter and setting the gradient so obtained to 0*, we get the desired result.*

Our negative results on the monotonicity of the probability density of a path with respect to variations in the kinetic parameters make it difficult to argue the monotonicity of the probability of a formula being true on a model. The definition of the unbiased statistical estimator is used to argue that its value remains monotonic in any given positive parameter space under a very mild technical condition. The only condition that we need to satisfy is that the estimator should not take the values 0 or 1 anywhere in the positive parameter space. This is indeed true for any non-trivial property of a realistic biochemical system.

Theorem 16 (Absence of Local Extrema of the Unbiased Statistical Estimator) *The unbiased statistical estimator of a non-trivial probability (true with probability neither* 0 *nor* 1*) of a measurable set estimated using a finite number of finite length paths in a stochastic biochemical model does not admit a local extrema anywhere in the positive kinetic parameter space.*

Proof 16 *Assume that the statistical estimator* $\hat{P}$ *does have a local extrema at* k_u *(*$\neq 0$*), for the sake of contradiction.*

$$\begin{aligned}\hat{P} &\overset{def}{\equiv} \frac{\sum_{L(\sigma_t)=1, 1\leqslant t\leqslant T} \prod_{i=1}^{l} k_{j_i}(\sigma_t)\gamma^{j_i}_{(i,i+1)}(\sigma_t)\exp(-\sum_{h=1}^{n} k_h(\sigma_t)\gamma^{h}_{(i,i+1)}(\sigma_t)\Delta_i(\sigma_t))}{\sum_{t=1}^{T} \prod_{i=1}^{l} k_{j_i}(\sigma_t)\gamma^{j_i}_{(i,i+1)}(\sigma_t)\exp(-\sum_{h=1}^{n} k_h(\sigma_t)\gamma^{h}_{(i,i+1)}(\sigma_t)\Delta_i(\sigma_t))} \\ &\overset{def}{\equiv} \frac{P_{one}(k_u)}{P_{all}(k_u)} \qquad (5.1)\end{aligned}$$

Note that $P_{one}(k_u)$ *represents the sum of the probability densities of all the sampled paths that satisfy the AFM specification* ϕ *and are labeled 1, and* $P_{all}(k_u)$ *simply represents the sum of the probability densities of all the sampled paths. Setting* $\frac{\partial \hat{P}}{\partial k_u}$ *to* 0*, we get:*

$$P_{all}(k_u)\frac{\partial P_{one}(k_u)}{\partial k_u} - P_{one}(k_u)\frac{\partial P_{all}(k_u)}{\partial k_u} = 0$$

$$\begin{aligned}\Rightarrow P_{all}(k_u)\frac{\partial P_{one}(k_u)}{\partial k_u} &= P_{one}(k_u)\frac{\partial P_{all}(k_u)}{\partial k_u} && \textit{Algebraic Manipulation} \quad (5.2)\\ \Rightarrow \frac{1}{P_{one}(k_u)}\frac{\partial P_{one}(k_u)}{\partial k_u} &= \frac{1}{P_{all}(k_u)}\frac{\partial P_{all}(k_u)}{\partial k_u} && \textit{Algebraic Manipulation} \quad (5.3)\\ \Rightarrow \frac{1}{\frac{P_{one}(k_u)}{P_{all}(k_u)}}\frac{\partial P_{one}(k_u)}{\partial k_u} &= \frac{\partial P_{all}(k_u)}{\partial k_u} && \textit{Algebraic Manipulation} \quad (5.4)\end{aligned}$$

$$\Rightarrow \frac{1}{\hat{P}(k_u)}\frac{\partial P_{one}(k_u)}{\partial k_u} \quad = \quad \frac{\partial P_{all}(k_u)}{\partial k_u} \qquad \textit{Definition of } \hat{P}(k_u) \tag{5.5}$$

Now, let $P_{zero}(k_u)$ represent the sum of the probability densities of all the sampled paths that do not satisfy the AFM specification ϕ.

$$\begin{aligned}
& P_{all}(k_u) = P_{one}(k_u) + P_{zero}(k_u) && \textit{By Definition} && (5.6)\\
\Rightarrow & \frac{\partial P_{all}(k_u)}{\partial k_u} = \frac{\partial P_{one}(k_u)}{\partial k_u} + \frac{\partial P_{zero}(k_u)}{\partial k_u} && \textit{Differentiating both sides} && (5.7)\\
\Rightarrow & \frac{1}{P_{all}(k_u)}\frac{\partial P_{all}(k_u)}{\partial k_u} = \frac{\frac{\partial P_{one}(k_u)}{\partial k_u} + \frac{\partial P_{zero}(k_u)}{\partial k_u}}{P_{one}(k_u) + P_{zero}(k_u)} && \textit{Dividing both sides by } P_{all}(k_u) && (5.8)
\end{aligned}$$

Also,

$$\begin{aligned}
& \frac{\frac{\partial P_{one}(k_u)}{\partial k_u}}{P_{one}} && = \frac{\frac{\partial P_{one}(k_u)}{\partial k_u} + \frac{\partial P_{zero}(k_u)}{\partial k_u}}{P_{one} + P_{zero}} && \textit{From 5.3 and 5.8} && (5.9)\\
\Rightarrow & \frac{\frac{\partial P_{one}(k_u)}{\partial k_u}}{P_{one}} && = \frac{\frac{\partial P_{zero}(k_u)}{\partial k_u}}{P_{zero}} && \textit{Algebraic Manipulation} && (5.10)\\
\Rightarrow & \frac{\frac{\partial P_{one}(k_u)}{\partial k_u}}{P_{one}/(P_{one} + P_{zero})} && = \frac{\frac{\partial P_{zero}(k_u)}{\partial k_u}}{P_{zero}/(P_{one} + P_{zero})} && \textit{Dividing both sides} && (5.11)\\
\Leftrightarrow & \frac{1}{\hat{P}(k_u)}\frac{\partial P_{one}(k_u)}{\partial k_u} && = \frac{1}{1 - \hat{P}(k_u)}\frac{\partial P_{zero}(k_u)}{\partial k_u} && \textit{Estimator Definition} && (5.12)
\end{aligned}$$

Hence,

$$\frac{1}{\hat{P}(k_u)}\frac{\partial P_{one}(k_u)}{\partial k_u} = \frac{1}{1 - \hat{P}(k_u)}\frac{\partial P_{zero}(k_u)}{\partial k_u} = \frac{\partial P_{all}(k_u)}{\partial k_u} \qquad \textit{From Eqn. 5.5 and Eqn. 5.12} \tag{5.13}$$

Now, given a finite set of paths S and the sum of probability density P_S of these paths,

$$P_S = \sum_{\sigma \in S} P(\sigma) \qquad \textit{(By Definition)} \tag{5.14}$$

$$\Rightarrow \frac{\delta P_S}{\delta k_u} = \frac{\delta}{\delta k_u} \sum_{\sigma \in S} P(\sigma) \tag{5.15}$$

$$\Rightarrow \frac{\delta P_S}{\delta k_u} = \sum_{\sigma \in S} \frac{\delta}{\delta k_u} \left(P(\sigma) \right) \qquad \textit{(Derivative of Finite Sums)} \tag{5.16}$$

$$\Rightarrow \frac{\delta P_S}{\delta k_u} = \sum_{\sigma \in S} \left(P(\sigma) \sum_{i=0}^{l(\sigma)-1} \left(\mathcal{I}(u = j_i(\sigma)) \frac{1}{k_u} - \mathcal{I}(r_u, i) \gamma^u_{(i,i+1)} \Delta_i(\sigma) \right) \right) \tag{5.17}$$

$$\Rightarrow \frac{\delta P_S}{\delta k_u} = \sum_{\sigma \in S} P(\sigma) \left(\frac{N_u(\sigma)}{k_u} - \sum_{i=0}^{l(\sigma)-1} \left(\mathcal{I}(r_u, i) \gamma^u_{(i,i+1)} \Delta_i(\sigma) \right) \right) \tag{5.18}$$

Now,

$$\tfrac{\partial P_{one}(k_u)}{\partial k_u} = \tfrac{P_{one}(k_u)}{P_{zero}(k_u)} \tfrac{\partial P_{zero}(k_u)}{\partial k_u} \qquad \textit{From Equation 5.10} \tag{5.19}$$

$$\Rightarrow \qquad \textstyle\sum_{\sigma \in S, \sigma \models \phi} P(\sigma) \left(\tfrac{N_u(\sigma)}{k_u} - \sum_{i=0}^{l(\sigma)-1} \mathcal{I}(r_u, i) \gamma^u_{(i,i+1)} \Delta_i(\sigma) \right)$$

$$= \textstyle\sum_{\sigma \in S, \sigma \models \phi} P(\sigma) \left(\tfrac{1}{P_{zero}(k_u)} \tfrac{\partial P_{zero}(k_u)}{\partial k_u} \right) \tag{5.20}$$

Consider a more accurate statistical estimator with one more sample s. Without loss of generality, assume that the sample satisfies the formula ϕ. Let k'_u be the point at which the new estimator reached an extrema.

$$\sum_{\sigma \in S \cup s, \sigma \models \phi} P(\sigma) \left(\frac{N_u(\sigma)}{k'_u} - \sum_{i=0}^{l(\sigma)-1} \mathcal{I}(r_u, i) \gamma^u_{(i,i+1)} \Delta_i(\sigma) \right) = \sum_{\sigma \in S \cup s, \sigma \models \phi} P(\sigma) \left(\frac{1}{P_{zero}(k'_u)} \frac{\partial P_{zero}(k'_u)}{\partial k_u} \right) \tag{5.21}$$

Thus, subtracting Eqn. 5.19 from Eqn. 5.21, we get

$$\sum_{\sigma\in S,\sigma\models\phi}\left(P(\sigma)\frac{N_u(\sigma)}{k'_u} - P(\sigma)\frac{N_u(\sigma)}{k_u}\right) + P(s)\left(\frac{N_u(s)}{k'_u} - \sum_{i=0}^{l(s)-1}\left(\mathcal{I}(r_u,i)\gamma^u_{(i,i+1)}\Delta_i(s)\right)\right)$$
$$= \sum_{\sigma\in S,\sigma\models\phi}\left(P(\sigma)\frac{1}{P_{zero}(k'_u)}\frac{\partial P_{zero}(k'_u)}{\partial k_u} - P(\sigma)\frac{1}{P_{zero}(k_u)}\frac{\partial P_{zero}(k_u)}{\partial k_u}\right) + P(s)\left(\frac{1}{P_{zero}(k'_u)}\frac{\partial P_{zero}(k'_u)}{\partial k_u}\right)$$

Since, $\frac{1}{k_u}$, P_{zero}, $P(\sigma)$, $\frac{\partial P_{zero}}{\partial k_u}$ are continuous functions of k_u in the positive parameter space, the following holds true as the number of samples used by the statistical estimator increases and $k'_u \to k_u$:

$$P(s)\left(\frac{N_u(s)}{k_u} - \sum_{i=0}^{l(s)-1}\left(\mathcal{I}(r_u,i)\gamma^u_{(i,i+1)}\Delta_i(s)\right)\right) = P(s)\left(\frac{1}{P_{zero}(k_u)}\frac{\partial P_{zero}(k_u)}{\partial k_u}\right)$$
$$\Rightarrow \quad k_u = \frac{N_u(s)}{\left(\frac{1}{P_{zero}(k_u)}\frac{\partial P_{zero}(k_u)}{\partial k_u} + \sum_{i=0}^{l(s)-1}\mathcal{I}(r_u,i)\gamma^u_{(i,i+1)}\Delta_i(s)\right)} \tag{5.22}$$

The location of the extrema is a function of the new sample we chose unless $\frac{1}{P_{zero}(k_u)}\frac{\partial P_{zero}(k_u)}{\partial k_u} \to \infty$. If $P_{zero}(k_u) \neq 0$, then the only possibility is $\frac{\partial P_{zero}(k_u)}{\partial k_u} \to \infty$ i.e. $\sum_{\sigma\in S} P(\sigma)(\frac{N_u(\sigma)}{k_u} - \sum_{i=0}^{l(\sigma)-1}(\mathcal{I}(r_u,i)\gamma^u_{(i,i+1)}\Delta_i(\sigma))) \to \infty$. The latter is not possible as $k_u \neq 0$ and the other terms are bounded for any simulation.

Thus, there is no extrema in the positive parameter space.

In this section, we have shown several important results about the probability of a formula being true on a stochastic biochemical model:

(i) The logarithm of the probability density of a path in a stochastic biochemical model is *uniformly* and *jointly* continuous in the logarithmic kinetic parameter space. Our proof is constructive and hence, suggests a natural sampling based algorithm which we present in Sec. 5.5.1. We are aware that a non-constructive proof would be simpler but not algorithmically useful.

(ii) The probability density of a path is not necessarily monotonic in the parameter space. Thus, the natural mechanism of using monotonicity of paths to argue the monotonicity of the statistical estimator of the probability of a model satisfying a formula is not available.

(iii) An indirect proof using *survey sampling* based unbiased statistical estimators establishes that the unbiased statistical estimator is indeed monotonic in the positive parameter space. This provides the opportunity for constructing efficient synthesis and search algorithms. We present these algorithms in Sec. 5.5.2 and Sec. 5.5.3 respectively.

5.5 Parameter Synthesis Algorithms

We have characterized the parameter space of an *adapted finitely monitorable* formula being true on a stochastic biochemical model in the previous section. Now, we use our understanding of the parameter space to suggest efficient algorithms for parameter synthesis of stochastic biochemical models against high-level behavioral specifications.

5.5.1 Algorithm 4: Parameter Synthesis using Uniform Continuity

We have shown that the probability density of a path does not change arbitrarily as we change the reaction rate parameters of a stochastic biochemical system. This result will now enable us to prove results on the *dense* parameter space (with uncountably many parameter values) by sampling *only finitely many parameter values* in the parameter space. Algorithm 4 takes five inputs:

(i) Stochastic Biochemical Model $\mathcal{M}$ with unknown kinetic parameters Θ,

(ii) A high-level behavioral specification about the system specified in a probabilistic adapted finitely monitorable logic $Pr_{\geqslant\rho}(\phi)$,

(iii) The space in which the possible values θ of reaction parameters are to be searched: $\theta \in [\Theta_1^{min}, \Theta_1^{max}] \times \cdots \times [\Theta_{n_\Theta}^{min}, \Theta_{n_\Theta}^{max}]$

(iv) An error tolerance η such that $\sqrt{\rho} < \eta < 1$: A number close to 1 which specifies the acceptable error in the synthesis of parameters. All points within the synthesized parameter set will satisfy the adapted finitely monitorable property with probability at least ρ, and those outside the set satisfy the specification with probability no greater than $\frac{\rho}{\eta^2}$.

(v) A confidence value, q, which will be passed to the statistical model validation algorithm that is called as a subroutine.

The algorithm initializes the set of satisfying parameters to the empty set. It then uses the error tolerance η to compute ϵ, the required resolution of the discretization of the logarithm of the probability space. Next, the algorithm discretizes the logarithmic parameter space. Note that the notation $[\log \Theta_1^{min}, \log \Theta_1^{max}]_\delta$ is used to represent the set

Algorithm 4 Parameter Synthesis using Statistical model validation

```
Require: Parameterized Biochemical Model $\mathcal{M}(\Theta)$,
         Probabilistic Adapted Finitely Monitorable Formula $Pr_{\geqslant\rho}(\phi)$,
         Parameter Space $\theta \in [\Theta_1^{min}, \Theta_1^{max}] \times \cdots \times [\Theta_{n_\Theta}^{min}, \Theta_{n_\Theta}^{max}]$,
         Error Tolerance in PAFM Specification $\sqrt{\rho} < \eta < 1$,
         Confidence Probability $q$.
Ensure: Set $\tilde{\mathcal{V}}_{\phi,\rho}$ of parameter values such that
         (i) $\forall\theta \in \tilde{\mathcal{V}}_{\phi,\rho},\ \mathcal{M}(\theta) \models_q Pr_{\geqslant\rho}(\phi)$, and
         (ii) $\forall\theta \notin \tilde{\mathcal{V}}_{\phi,\rho},\ \mathcal{M}(\theta) \models_q Pr_{\leqslant\frac{\rho}{\eta^2}}(\phi)$.

  // Initialize set of parameter values to the empty set.
  $\tilde{\mathcal{V}}_{\phi,\rho} = \{\}$
  // Compute $\epsilon$ and $\delta$ from $\eta$.
  $\epsilon = |\log\eta|$
  $\delta = \delta(\epsilon/2, \mathcal{M})$
  // Search the discretized parameter space.
  for all $\theta_{log} \in [\log\Theta_1^{min}, \log\Theta_1^{max}]_\delta \times \cdots \times [\log\Theta_{n_\Theta}^{min}, \log\Theta_{n_\Theta}^{max}]_\delta$ do
    // If a parameter value satisfies the PAFM formula with probability $\frac{\rho}{\eta}$
    if $\mathcal{M}(\exp(\theta_{log})) \models_q Pr_{\geqslant\frac{\rho}{\eta}}(\phi)$ then
      // Add the $\delta$-ball around this parameter value to the set $S$.
      $\tilde{\mathcal{V}}_{\phi,\rho} = \tilde{\mathcal{V}}_{\phi,\rho} \cup \{\bigcup_{b\in\mathcal{B}_\delta(\theta_{log})}\{\exp(b)\}\}$    // $B_\delta(x) \overset{def}{=} \{y \mid |x-y| \leqslant \delta\}$
    end if
  end for
  if $\tilde{\mathcal{V}}_{\phi,\rho} == \{\}$ then
    Print “Model is infeasible".
  end if
```

$\{\log\theta \mid \log\Theta_1^{min} \leqslant \log\theta \leqslant \log\Theta_1^{max}$, and $\log\theta = z.\delta$ for some $z \in \mathbb{Z}\}$. For each discrete box in the logarithmic parameter space, the algorithm samples a point and tests whether it satisfies the property with probability at least $\frac{\rho}{\eta}$. If so, the algorithm adds the exponential of this point (and the discrete hyperbox of size δ around it) to the set of synthesized parameters. We use the notation $\mathcal{B}_\delta((c_1, \ldots c_n))$ to represent the hyperbox of size δ around the center point $(c_1, \ldots c_n)$ i.e. the set $\{(x_1, \ldots x_n) \mid \max_{1 \leqslant i \leqslant n} |c_i - x_i| \leqslant \delta\}$. The algorithm terminates after each discrete box has been examined. If no parameter combination produces a model that satisfies the formula, the algorithm reports that the model is *infeasible* with respect to the given high-level behavioral specification. Knowledge about the infeasibility of a model is of practical importance, because it indicates that the model itself has structural flaws (eg., missing biochemical pathways) which need to be addressed before parameter synthesis can be attempted. This is really significant because manual *ad hoc* search procedures can never prove the infeasibility of the model with respect to the behavioral specification, and the designer is left to wonder if the model is actually infeasible or she has just not found the right parameters for the model yet.

Theorem 17 *If θ is a point in the synthesized parameter set $\tilde{\mathcal{V}}_{\phi,\rho}$ returned by Algorithm 4, $\mathcal{M}(\theta) \models_q Pr_{\geqslant\rho}(\phi)$.*

Proof 17 *Suppose θ is a point in $\tilde{\mathcal{V}}_{\phi,\rho}$. Then,*

(i) By the construction of the set $\tilde{\mathcal{V}}_{\phi,\rho}$ by the algorithm, there exists a θ_{log} such that $\theta \in \bigcup_{b\in\mathcal{B}_\delta(\theta_{log})}\{\exp(b)\}$ and $\mathcal{M}(\exp(\theta_{log})) \models_q Pr_{\geqslant\frac{\rho}{\eta}}(\phi)$.

(ii) Since $\theta \in \bigcup_{b\in\mathcal{B}_\delta(\theta_{log})}\{\exp(b)\}$ i.e. θ lies in the δ-neighbourhood of θ_{log}, $|\theta - \theta_{log}| < \delta$.

(iii) By uniform continuity, we know that $\mathcal{M}(\theta)$ satisfies $Pr_{\geqslant\left(\frac{\rho}{\eta}e^{\epsilon}\right)}(\phi)$. Note that $e^{\epsilon} < 1$

by construction.

(iv) By our choice of η, we know that $e^{\epsilon} = \eta$.

Hence, $\mathcal{M}(\theta)$ satisfies $Pr_{\geqslant\rho}(\phi)$ up to the confidence probability q.

Theorem 18 *If θ is not a point in the synthesized parameter set $\tilde{\mathcal{V}}_{\phi,\rho}$ returned by Algorithm 4, $\mathcal{M}(\theta)$ satisfies $Pr_{\leqslant\frac{\rho}{\eta^2}}(\phi)$.*

Proof 18 *Suppose θ is not a point in $\tilde{\mathcal{V}}_{\phi,\rho}$. Then,*

(i) By the construction of the set $\tilde{\mathcal{V}}_{\phi,\rho}$ in the algorithm, there does not exists any point θ_{log} such that $\theta \in \bigcup_{b\in\mathcal{B}_\delta(\theta_{log})}\{\exp(b)\}$ and $\mathcal{M}(\exp(\theta_{log})) \models_q Pr_{\geqslant\frac{\rho}{\eta}}(\phi)$.

(ii) But the algorithm must have sampled a point θ'_{log} such that $\theta \in \bigcup_{b\in\mathcal{B}_\delta(\theta'_{log})}\{\exp(b)\}$.

(iii) Since the algorithm did not add this point to S, it must be the case that $\mathcal{M}(\exp(\theta'_{log})) \models_q Pr_{<\frac{\rho}{\eta}}(\phi)$.

(iv) Since $\theta \in \bigcup_{b\in\mathcal{B}_\delta(\theta'_{log})}\{\exp(b)\}$ i.e. θ lies in the δ-neighbourhood of θ'_{log}, $|\theta - \theta'_{log}| < \delta$.

(v) By uniform continuity, we know that $\mathcal{M}(\theta)$ satisfies $Pr_{\leqslant(\frac{\rho}{\eta\; e^{\epsilon}})}(\phi)$.

(vi) By our choice of η, we know that $e^{\epsilon} = \eta$.

Hence, $\mathcal{M}(\theta)$ satisfies $Pr_{\leqslant\frac{\rho}{\eta^2}}(\phi)$ up to the confidence probability q.

Theorem 19 *The number of discrete parameter values sampled by the algorithm is polylogarithmic in the error tolerance of the PAFM specification η.*

Proof 19 *Given the error tolerance η, the discretization ϵ chosen by the algorithm in the logarithmic probability space is logarithmic in the error tolerance η. The discretization δ of the logarithmic parameter space is $C\epsilon$, where C is a factor independent of ϵ.*

From the algorithm, we know that the number of parameter values to be sampled is $\left(\max_{1 \leqslant i \leqslant n_\Theta} n_\Theta \log \frac{\Theta_i^{max}}{\Theta_i^{min}} \frac{1}{\delta}\right)^{n_\Theta}$*. Rewriting, the number of sampled values is* $\left(\max_{1 \leqslant i \leqslant n_\Theta} \frac{n_\Theta}{\delta^{n_\Theta}} \log \frac{\Theta_i^{max}}{\Theta_i^{min}}\right)$*, which is the same as* $\left(\max_{1 \leqslant i \leqslant n_\Theta} \frac{n_\Theta}{(C|\log \eta|)^{n_\Theta}} \log \frac{\Theta_i^{max}}{\Theta_i^{min}}\right)$.

5.5.2 Algorithm 5: Faster Parameter Synthesis using Abstraction Refinement

We have also shown that the unbiased statistical estimator for the probability of a measurable set of paths satisfying a formula does not admit any local extrema as we change the reaction rate parameters, unless the probability of a formula being true on the model is either unity or zero somewhere in the logarithmic parameter space being explored. We can therefore modify the previous algorithm to perform a hierarchical search through the parameter space to accelerate synthesis. We refer to this hierarchical decomposition of the parameter space as *abstraction-refinement*. We are not the first to use hierarchical abstractions and binary search refinements. Similar ideas have been pursued in the context of nonlinear hybrid systems [JBS07] and stochastic systems [HKM08]. However, our monotonicity results (see Section 5.4) considerably simplify our algorithm.

The central idea behind abstraction refinement is illustrated in Fig. 5.2. Suppose we want to check that a certain formula is true with probability *no more* than 0.14. If all the corners of a hyperbox satisfy the formula, or if all the corners do not satisfy the satisfy the formula, then our monotonicity results (see Sec. 5.4) allows one to stop further analysis of this hyperbox. In the figure, the lower right corner box clearly does not satisfy the PAFM specification and need not be further analyzed at all.

Algorithm 5 Faster Parameter Synthesis using Abstraction Refinement

```
Require: Parameterized Biochemical Model $\mathcal{M}(\Theta)$, Probabilistic Adapted Finitely Monitorable Formula $Pr_{\geqslant\rho}(\phi)$, Parameter Space $\theta \in [\Theta_1^{min}, \Theta_1^{max}] \times \cdots \times [\Theta_{n_\Theta}^{min}, \Theta_{n_\Theta}^{max}]$, Error Tolerance in PAFM Formula $\sqrt{\rho} < \eta < 1$, Confidence Probability $q$.
Ensure: Set $\tilde{\mathcal{V}}_{\phi,\rho}$ of parameter values such that
        (i) $\forall\theta \in \tilde{\mathcal{V}}_{\phi,\rho},\ \mathcal{M}(\theta) \ \models_q \ Pr_{\geqslant\rho}(\phi)$, and
        (ii) $\forall\theta \in \tilde{\mathcal{V}}^c_{\phi,\rho},\ \mathcal{M}(\theta) \ \models_q \ Pr_{<\rho}(\phi)$.
  $\tilde{\mathcal{V}}_{\phi,\rho} = \{\}$, $\tilde{\mathcal{V}}^c_{\phi,\rho} = \{\}$ {Initialize satisfying (unsatisfying) parameter values}
  $\epsilon = |\log\eta|$ {Compute discretization constant $\epsilon$ from error tolerance $\eta$}
  for all $i$ = 1 to $n_C$ do
    $\delta_i = \frac{C_i^{max} - C_i^{min}}{2}$
  end for
  $U = [\log C_1^{min}, \log C_1^{max}]_{\delta_1} \times \cdots \times [\log C_{n_C}^{min}, \log C_{n_C}^{max}]_{\delta_{n_C}}$ {Search discretized space}
  for all $\theta_{log} \in U$ do
    {$H(x)$ is a hyperbox with each side of length $\delta_i$ around $x$.}
    if $\underline{\mathcal{M}}(H(\theta_{log})) \ \models_q \ Pr_{\geqslant\rho}(\phi)$ then
      $\tilde{\mathcal{V}}_{\phi,\rho} = \tilde{\mathcal{V}}_{\phi,\rho} \cup \{exp(H(\theta_{log}))\}$ {Parameter value satisfies the spec. with probability at least $\rho$} {Add hyperbox around this parameter value to $S$.}
    end if
    if $\overline{\mathcal{M}}(H(\theta_{log})) \ \models_q \ Pr_{<\rho}(\phi)$ then
      $\tilde{\mathcal{V}}^c_{\phi,\rho} = \tilde{\mathcal{V}}^c_{\phi,\rho} \cup \{exp(H(\Theta_{log}))\}$ {Parameter value satisfies the spec. with probability less than $\rho$.} {Add hyperbox around this parameter value to $S^c$.}
    end if
    $U = U \setminus (\tilde{\mathcal{V}}_{\phi,\rho} \cup \tilde{\mathcal{V}}^c_{\phi,\rho})$ {Update the unknown part of the parameter space.}
    for all $i$ = 1 to $n_C$ do
      $\delta_i = \frac{\delta_i}{2}$
    end for
    if $\delta_i < \delta(\epsilon/2, M)$ then
      Report these hyperboxes as unknown. {Hyperbox Size is too small}
      Break;
    end if
    $U = [\log C_1^{min}, \log C_1^{max}]_{\delta_1} \times \cdots \times [\log C_{n_C}^{min}, \log C_{n_C}^{max}]_{\delta_{n_C}}$ {Further discretize the unknown part of the parameter space.}
  end for
  if S == { } then
    Print “Model is infeasible".
  end if
```

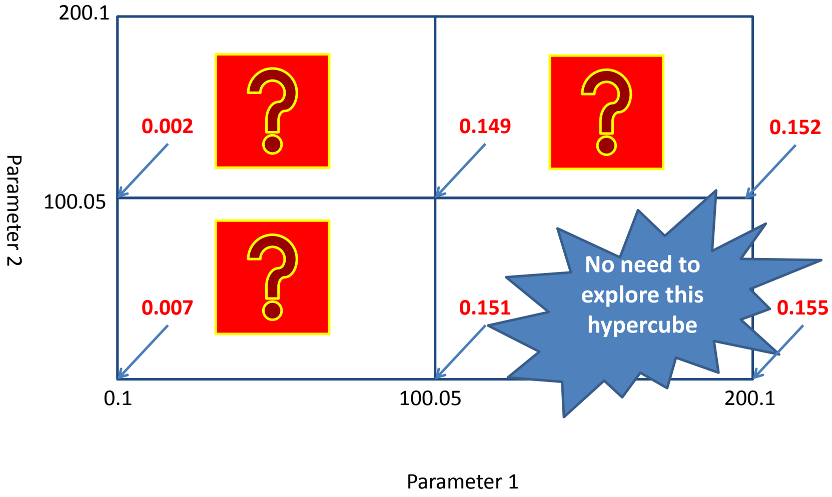


Figure 5.2: Central Idea behind the Abstraction Refinement Algorithm

The abstraction refinement based algorithm takes the same parameters as our earlier algorithm. However, it assumes that the probability of the formula does not vanish or reach unity anywhere in the parameter space being analyzed.

Given the monotonicity of the probability in each of the parameters, we can construct underapproximate and overapproximate abstractions in a bounded parameter space. An underapproximate (overapproximate) abstraction is the parameterized model which satisfies an AFM specification with the minimum (maximum) probability in a given parameter space.

Definition 18 (Underapproximate Abstraction) *Given parameters Θ and a bounded parameter space $S = [\Theta_1^{min}, \Theta_1^{max}] \times \cdots \times [\Theta_{n_\Theta}^{min}, \Theta_{n_\Theta}^{max}]$ over which the probability of the*

AFM specification being true is monotonic in each of the parameters, the variable m_i is assigned the value 1 if the probability of a formula being true is monotonically increasing in the parameter Θ_i; it is zero if the probability is monotonically decreasing.

The underapproximate abstraction of the set of models $\mathcal{M}(\theta)$ in the bounded parameter space $\theta \in [\Theta_1^{min}, \Theta_1^{max}] \times \cdots \times [\Theta_{n_\Theta}^{min}, \Theta_{n_\Theta}^{max}]$ is given by the model $\mathcal{M}(\theta_{min})$, where $\theta_{min} = \big((1 - m_1)(\Theta_1^{max} - \Theta_1^{min}) + \Theta_1^{min}, \ldots, (1 - m_{n_\Theta})(\Theta_{n_\Theta}^{max} - \Theta_{n_\Theta}^{min}) + \Theta_{n_\Theta}^{min}\big)$. We also denote the underapproximate model $\mathcal{M}(\theta_{min})$ by $\underline{\mathcal{M}}(S)$.

Definition 19 (Overapproximate Abstraction) *Given parameters Θ and a bounded parameter space $[\Theta_1^{min}, \Theta_1^{max}] \times \cdots \times [\Theta_{n_\Theta}^{min}, \Theta_{n_\Theta}^{max}]$ over which the probability of the AFM specification being true is monotonic in each of the parameters, the variable m_i is assigned the value 1 if the probability of a formula being true is monotonically increasing in the parameter Θ_i; it is zero if the probability is monotonically decreasing.*

The overapproximate abstraction of the set of models $\mathcal{M}(\theta)$ in the bounded parameter space $\theta \in [\Theta_1^{min}, \Theta_1^{max}] \times \cdots \times [\Theta_{n_\Theta}^{min}, \Theta_{n_\Theta}^{max}]$ is given by the model $\mathcal{M}(\theta_{min})$, where $\theta_{min} = \big(m_1(\Theta_1^{max} - \Theta_1^{min}) + \Theta_1^{min}, \ldots, m_{n_\Theta}(\Theta_{n_\Theta}^{max} - \Theta_{n_\Theta}^{min}) + \Theta_{n_\Theta}^{min}\big)$. We also denote the overapproximate model $\mathcal{M}(\theta_{min})$ by $\overline{\mathcal{M}}(S)$.

The algorithm first constructs two empty sets $\tilde{\mathcal{V}}_{\phi,\rho}$ and $\tilde{\mathcal{V}}^c_{\phi,\rho}$ containing the space of parameters that do and do not satisfy the specification, respectively. We also compute a coarse-grained discretization of the parameter space by dividing each parameter value into two parts. For each hyperbox formed, if the model satisfies the probabilistic adapted finitely monitorable logic formula with probability more than ρ for the underapproximate model $\underline{\mathcal{M}}$ in the parameter space defined by the hyperbox, then we add this hyperbox to the set $\tilde{\mathcal{V}}_{\phi,\rho}$ of parameter values satisfying the specification. The monotonicity of the parameter values (See Theorem 16) provide the technical justification for doing so. On the contrary, if the model satisfies the formula with probability less than ρ for the overapproximate model $\overline{\mathcal{M}}$ in the parameter space defined by the hyperbox, then we add this hyperbox to the set $\tilde{\mathcal{V}}^{c}_{\phi,\rho}$ of parameter values not satisfying the specification.

If a hyperbox is neither in the set of parameter values satisfying or not satisfying the PAFM specification, and is larger than the minimal threshold size dictated by the error tolerance of the PAFM specification η, we refine the hyperbox by splitting each parameter value into two parts and we continue with the algorithm. If the hyperbox has become smaller than the threshold size dictated by the error tolerance η, we stop analyzing this hyperbox any further.

Theorem 20 *If θ is a point in the set of unknown hyperboxes U in Algorithm 5 with error tolerance η, then $\mathcal{M}(\theta) \models_q Pr_{<\frac{\rho}{\eta}}(\phi)$, and $\mathcal{M}(\theta) \models_q Pr_{>\rho\eta}(\phi)$.*

Proof 20 *We will show that $\mathcal{M}(\theta) \models_q Pr_{<\frac{\rho}{\eta}}(\phi)$, and $\mathcal{M}(\theta) \models_q Pr_{>\rho\eta}(\phi)$ if $\theta \in U$.*

(i) Suppose θ satisfies $\mathcal{M}(\theta) \models_q Pr_{\geqslant\frac{\rho}{\eta}}(\phi)$. Consider the hyperbox $H(\theta)$ of size $\delta(\epsilon, M)$ around this point. By uniform continuity and our choice of discretization, every point c in the hyperbox (in particular, the corners of the hyperbox) satisfy $\mathcal{M}(c) \models_q Pr_{\geqslant\frac{\rho}{\eta}e^{\epsilon}}(\phi)$. But, by construction, we know that $e^{\epsilon} = \eta$. Hence, the corners satisfy the formula $Pr_{\geqslant\rho}(\phi)$ with confidence q. But, in that case, θ would be in $\tilde{\mathcal{V}}_{\phi,\rho}$ and not in U.

(ii) Suppose that θ satisfies $\mathcal{M}(\theta) \models_q Pr_{\leqslant\rho\eta}(\phi)$. Consider the hyperbox $H(\theta)$ of size $\delta(\epsilon, M)$ around this point. By uniform continuity, every point c in the hyperbox

(in particular, the corners of the hyperbox) satisfy $\mathcal{M}(c) \models_q Pr_{\leqslant \frac{\rho\eta}{e^{\epsilon}}}(\phi)$. But, by construction, we know that $e^{\epsilon} = \eta$. But, in that case, $\mathcal{M}(c) \models_q Pr_{\leqslant \rho}(\phi)$ and θ would be in $\tilde{\mathcal{V}}^{c}_{\phi,\rho}$ and not in U.

The theorem points out that the quality of the answer obtained by our abstraction refinement algorithm depends on the error tolerance parameter η. As η approaches one, the size of the set U with unknown parameter values becomes smaller.

5.5.3 Algorithm 6: Parameter Search and Model Infeasibility using Gradient Descent

The previous two algorithms solve the parameter synthesis problem. In this section, we consider a slightly different problem, and find the parameter combination that maximizes the probability that the given formula will hold on the model, or reports that the model and the PAFM specification are mutually infeasible over the given parameter space. The algorithm takes the same inputs as the parameter synthesis algorithms. It begins by computing the smallest step size, δ, that will guarantee that the ratio of the probabilities associated with any two points inside any hyperbox of length δ will not exceed η. Then, the algorithm samples a random point in the parameter space and computes the gradient of the probability at this point in the parameter space using the equations in Theorem 4. The algorithm then moves by a step of δ in the direction of the gradient. If the algorithm crosses the parameter space to be searched, it stops. The algorithm checks if the last point sampled in the parameter space satisfies the adapted finitely monitorable property with probability at least ρ. If so, it reports this point in the parameter space as the best parameter that can be synthesized. Otherwise, it declares that the parameter space does not contain any parameter values that enable the model to η-*robustly* satisfy the *PAFM* specification i.e. there exists no parameter value θ such that $\mathcal{M}(\theta) \ \models_q \ Pr_{\geqslant \frac{\rho}{\eta}}(\phi)$.

Algorithm 6 Synthesis and Infeasibility Analysis using Gradient Descent

```
Require:  Parameterized Biochemical Model $\mathcal{M}(\Theta)$,
          Probabilistic Adapted Finitely Monitorable Formula $Pr_{\geqslant\rho}(\phi)$,
          Parameter Space $\theta \in [\Theta_1^{min}, \Theta_1^{max}] \times \cdots \times [\Theta_{n_\Theta}^{min}, \Theta_{n_\Theta}^{max}]$,
          Error Tolerance in PAFM Formula $\sqrt{\rho} < \eta < 1$,
          Confidence Probability $q$,
Ensure:
          (i)  A point $\theta_0$ of parameter values such that $\mathcal{M}(\theta_0) \models_q Pr_{\geqslant\rho}(\phi)$, or
          (ii) Show that for all $\theta$ in the parameter space $\mathcal{M}(\theta) \models_q Pr_{<\frac{\rho}{\eta}}(\phi)$
{ Initialize parameter value to a random point}
$\theta_0 = RandomPoint([\Theta_1^{min}, \Theta_1^{max}] \times \cdots \times [\Theta_{n_\Theta}^{min}, \Theta_{n_\Theta}^{max}])$
{Compute $\epsilon$ and $\delta$ from $\eta$}
$\epsilon = |\log \eta|$
$\delta = \delta(\epsilon/2, \mathcal{M})$
{Search the discretized parameter space}
while $\theta_0 \in [\log \Theta_1^{min}, \log \Theta_1^{max}]_\delta \times \cdots \times [\log \Theta_{n_\Theta}^{min}, \log \Theta_{n_\Theta}^{max}]_\delta$ do
    $Gradient(\theta_0) = \frac{\delta P}{\delta \Theta}$, where $\mathcal{M}(\exp(\theta_0)) \ \models_q \ Pr_{=P}(\phi)$
    $\theta_0 = \theta_0 + \frac{Gradient(\theta_0)}{|Gradient(\theta_0)|}\delta$
end while
{If the parameter value satisfies the PAFM formula with probability $\rho$}
if $\mathcal{M}(\exp(\theta_0)) \ \models_q \ Pr_{\geqslant\rho}(\phi)$ then
    Print parameter $\exp(\theta_0)$. {Report this parameter value.}
    STOP.
end if
Print “Model is infeasible".
```

Theorem 21 *If the algorithm reports that the model is infeasible, then there does not exist any parameter value θ in the specified parameter space such that $\mathcal{M}(\theta) \models_q Pr_{\geqslant \frac{\rho}{\eta}}(\phi)$.*

Proof 21 *Suppose the algorithm reports that the model is infeasible and there is actually a point θ such that $\mathcal{M}(\theta) \models_q Pr_{\geqslant \frac{\rho}{\eta}}(\phi)$. Consider the $\delta(\epsilon, \mathcal{M})$-neighborhood of θ. If our algorithm sampled a point θ' in this neighborhood, it would have found that $\mathcal{M}(\theta') \models_q Pr_{\geqslant \rho}(\phi)$ (from Theorem 12) and stopped.*

Hence, our algorithm must not have sampled a point in the $\delta(\epsilon, \mathcal{M})$-neighborhood of θ. As the probability of a formula being true is monotonic over the parameter space, this is only possible if our algorithm sampled a parameter value with a higher probability that the probability associated with $\delta(\epsilon, \mathcal{M})$-neighborhood of θ. Thus, the algorithm could not have reported that the model is infeasible, contradicting our assertion.

Lemma 2 *If the algorithm reports that θ as the best synthesized parameter value with probability ρ_{syn} and θ_{max} is the actual parameter value that satisfies the specification with highest probability ρ_{max}, then $\rho_{syn} \geqslant \eta\rho_{max}$.*

Proof 22 *Suppose the algorithm reports ρ_{syn} such that $\rho_{syn} < \eta\rho_{max}$. But, θ is in the $\delta(\epsilon, \mathcal{M})$-neighborhood of θ_{max}; otherwise, the algorithm would not have stopped. By invoking uniform continuity (from Theorem 12), we have a proof by contradiction.*

Theorem 22 *The number of parameter points that the algorithm may need is bounded by*

Proof 23 *Given a monotonic parameter space $[\Theta_1^{min}, \Theta_1^{max}] \times \cdots \times [\Theta_{n_\Theta}^{min}, \Theta_{n_\Theta}^{max}]$, the algorithm moves only along the direction of increasing probability of the specification being true for each dimension or probability. The total number of discrete points along the i^{th} dimension is $\lceil \frac{\Theta_i^{max} - \Theta_i^{min}}{\delta} \rceil$. Hence, the total number of points that the algorithm may need to analyze is bounded by $\sum_{1 \leqslant i \leqslant n_\Theta} \lceil \frac{\Theta_i^{max} - \Theta_i^{min}}{\delta} \rceil$.*

5.6 Experimental Results

We analyzed stochastic models [HKN$^+$06, HKN$^+$08] of the Fibroblast Growth Factor and Cell Cycle biochemical signalling pathways against known behavioral specifications. Fibroblast growth factors (FGF) are a family of molecules involved in embryonic development, healing of wounds, and the development of new blood vessels (angiogenesis). As FGFs control the growth and differentiation of cells and are involved in angiogenesis, perturbations of this pathway are relevant at various stages of the development of cancer.

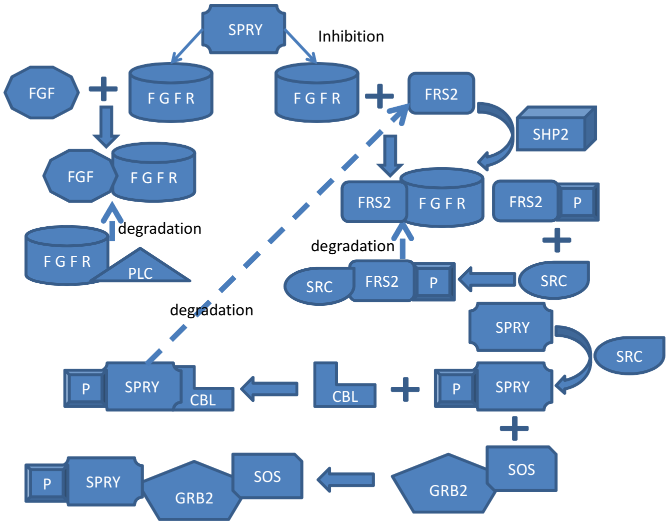


Figure 5.3: Cartoon Representation of the Fibroblast Growth Factor Receptor Pathway.

The FGF model (Fig. 5.3) comprises 10 base species: (i) the FGF molecule; (ii) the FGF receptor (FGFR); (iii) a FGFR-specific substrate (FRS2); (iv) the phosphatase Shp2; (v-vi) the kinases PLC and Src; (vii) the inhibitor Spry; (viii) the ubiquitin ligase Cbl; (ix) the adaptor protein Grb; and (x) the exchange factor Sos. These base species can bind to form additional species (FGF-FGFR, FGFR-FRS2, Shp2-FRS2, Src-FRS2, Grb-FRS2, PLC-FGFR,Spry-Src, Spry-Cbl,Spry-Grb, Grb-Sos), or degrade. The phosphatase and kinases cause state changes in the species (i.e., dephosphorylation and phosphorylation, respectively). The binding of Grb to Sos is an important event in the MAPK/ERK pathway, which regulates translation and transcription in the cell.

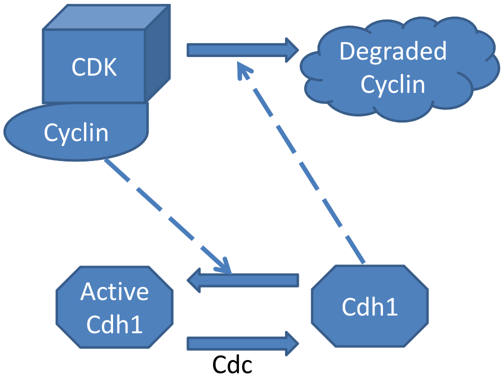


Figure 5.4: Cartoon Representation of the Cell Cyle Pathway
(Dashed arrows indicate catalysis/promotion of a reaction by a substrate)

We also studied a model of cell cycle control (Fig. 5.4) that consists of five species:

(i) cyclin; (ii) cyclin-dependent protein kinases (CDKs); (iii) cyclin-dependent kinase inhibitor (CKI); (iv) the phosphatase Cdc; and (v) the tumor suppressor gene Cdh1. CDKs are activated by binding to cyclins, and they control DNA synthesis and chromosome condensation during the initial phases of cell division. CDK activity is regulated through multiple mechanisms including cyclin, CKIs, and phosphorylation. We studied the absence of bound cyclin in our properties and the influence of kinetic parameters on the binding of cyclin.

5.6.1 Parameter Synthesis using Uniform Continuity (Algorithm 4)

We first performed experiments using the simplest of our algorithms discussed in Section 5.5.1. For the Fibroblast Growth Factor model, we perform parameter synthesis on a high level behavioral specification expressed in Probabilistic Bounded Metric Temporal Logic that was studied in [HKN$^+$06, HKN$^+$08]. The behavior concerns the probability that molecule Grb2 is bound to molecule FRS2 (denoted by $FRS2\_GRB$), and that FRS2 is not degraded at the time instant T.

In Fig. 5.5, we studied the influence of varying the FGF-FGFR dissociation parameter on the probability of the formula being true. We show the results of using our algorithms on the following formula:

$$Pr_{\geqslant 3.0\times 10^{-4}}\ [\ \mathbf{True}\ \mathbf{U}^{[1,1]}(FRS2\_GRB > 0\ \&\ degFRS2 = 0)\ ]$$

Our synthesized parameter space is correct with probability 0.99 (i.e., $q = 0.99$ in Al-

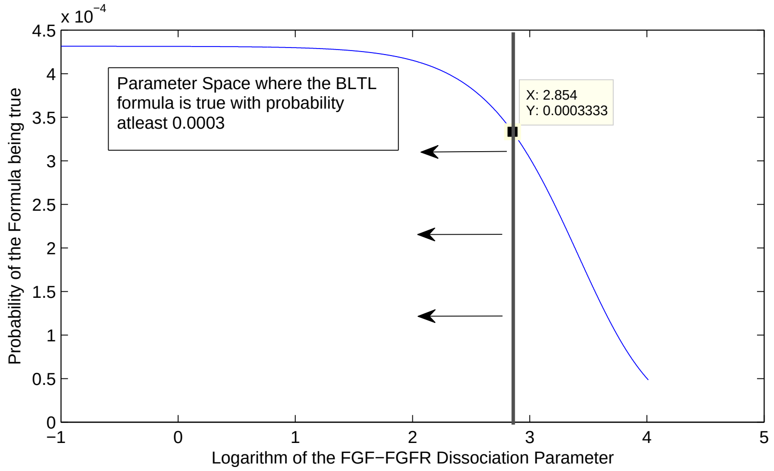


Figure 5.5: Synthesized 1-D Parameter Space
The parameter space to the left of the arrows satisfies the formula

$$Pr_{\geqslant 3.0\times 10^{-4}}\ [\ \mathbf{True}\ \mathbf{U}^{[1,1]}(FRS2\_GRB > 0\ \&\ degFRS2 = 0)\ ].$$

gorithm 4). We are able to demonstrate that the PBMTL formula is true whenever the *logarithm* of the FGF-FGFR dissociation parameter lies between $-1\ s^{-1}$ and $2.854\ s^{-1}$.

We considered the problem of synthesizing two parameters simultaneously — the Spry-SRC and FGF-FGFR dissociation rates, using the PBLTL formula stated above. The results of our experiments are plotted in Fig. 5.6. The region of the plot to the right of the contour line denoting probability 0.00033 contains they only parameter values that enable the model to satisfy the PBLTL formula. Note that the reason the value is 0.00033, and not 0.0003, is because we chose 0.91 as our choice of η in Algorithm 4. Hence, the parameter space we can confidently state to satisfy the PAFM formula based on our discretized sampling algorithm is the one which satisfies the formula with probability at least $\frac{0.0003}{0.91} = 0.00033$.

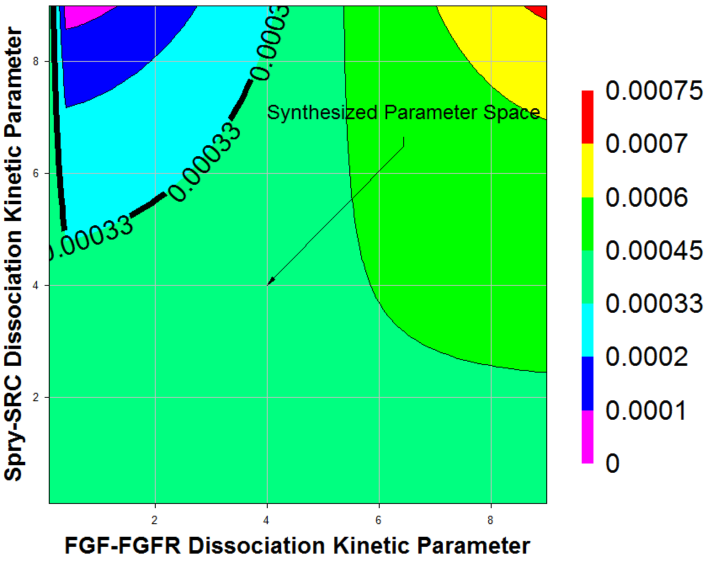


Figure 5.6: Synthesized 2-D Parameter Space
The synthesized parameter space lying to the right of the contour 0.00033 satisfies
$Pr_{\geqslant 3.0\times 10^{-4}}$ [**True U**$^{[1,1]}(FRS2\_GRB > 0 \ \& \ degFRS2 = 0)$]

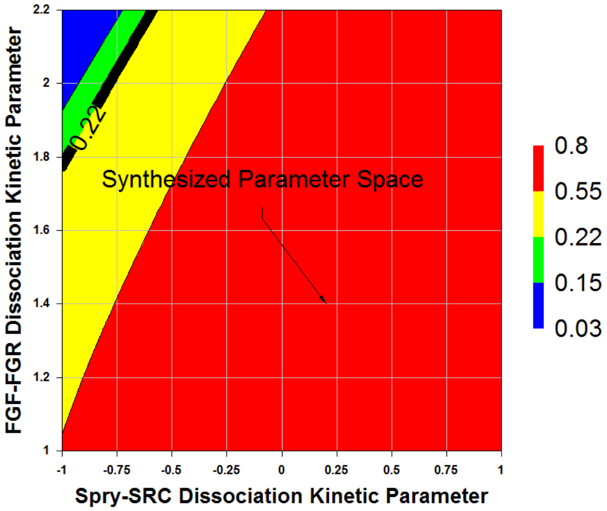


Figure 5.7: Synthesized Parameter Space for Fibroblast Growth Factor model
The synthesized parameter space lying to the right of the contour 0.22 satisfies the formula $Pr_{\geqslant 0.2}$ [**True U**$^{[60,60]}(FRS2\_GRB > 0 \ \& \ degFRS2 = 0)$]

We also analyzed the following PBMTL property:

$$Pr_{\geqslant 0.2} \ [\ \mathbf{True}\ \mathbf{U}^{[60,60]}(FRS2\_GRB > 0\ \&\ degFRS2 = 0)\ ]$$

Note that the two differences between this and the previous property are the time and probability bounds. The results obtained by our analysis are shown in Fig. 5.7. The algorithm reports that the combination of parameters to the right of the contour labeled 0.22 satisfy the formula.

Next, we analyzed the Cell Cycle model using uniform continuity arguments against the following Probabilistic Bounded Linear Temporal Logic property:

$$Pr_{\geqslant 0.4} \ [\ \mathbf{True}\ \mathbf{U}^{60} cyclin\_bound = 0]$$

The results of our analysis are presented in Fig. 5.6.1. The algorithm reports that the combination of parameters below the contour labeled 0.22 satisfy the formula.

Algorithm 4 is computationally quite expensive; it required about three days on a forty node cluster to run our experiments for a two dimensional system. The search for algorithms with low computational requirements yields the abstraction refinement algorithm that we present in the next section. The abstraction refinement based synthesis algorithms are more scalable and we discuss an benchmark example with six dimensions.

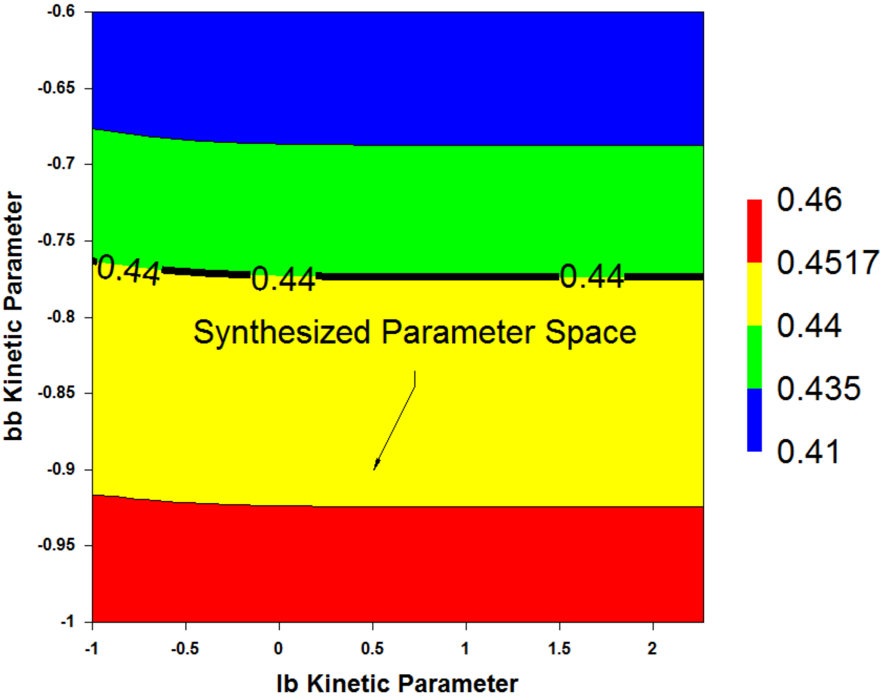


Figure 5.8: Synthesized Parameter Space for the Cell Cycle Model
The synthesized parameter space lying below the contour 0.44 satisfies

$$Pr_{\geqslant 0.4}\ [\ \mathbf{True}\ \mathbf{U}^{60} cyclin\_bound = 0]$$

5.6.2 Parameter Synthesis using Abstraction Refinement (Algorithm 5)

In this section, we report the results of using our abstraction refinement based algorithm (see Algorithm 5). We studied the performance of building abstractions using the FGF signal transduction pathway model (see Fig. 5.3) with as many as six parameters. We first asked our algorithm to synthesize the parameter where following statement is true:

$$Pr_{\geqslant 0.8}\ [\ \textbf{True}\ \mathbf{U}^{[60,60]}(FRS2\_GRB > 0\ \&\ degFRS2 = 0)\ ]$$

Our implementation took 30.7 minutes and confirmed that it is infeasible for the model to satisfy the formula in the parameter range being searched. Naturally, we cannot visualize the surface in a 6-dimensional space, as we could for the 1 and 2 dimensional cases. We note that our uniform continuity based algorithms would not be able to answer this question within a reasonable amount of time. For example, assuming a discretization of 0.1 in the logarithmic parameter space, it would take over 500 years for the proof to be completed. Thus, monotonicity is a really important property for tackling problems in high dimensions.

In Table 5.1, we used the following property for synthesizing the parameter space:

$$Pr_{\geqslant 0.15}\ [\ \textbf{True}\ \mathbf{U}^{[1,1]}(FRS2\_GRB > 0\ \&\ degFRS2 = 0)\ ]$$

Note that the number of hyperboxes to be explored and the time taken for the exhaus-

#Hyperboxes (Exhaustive Search)	Time (Exhaustive Search)	#Hyperboxes (Algorithm 5)	Time (Algorithm 5)
64	27.46 min.	64	27.46 min.
4,096	9.8 hours	640	1.55 hours
262,144	13.9 days*	15,744	20.1 hours
16,777,216	1.7 years*	<1,007,616*	<38 days*

Table 5.1: Abstraction Refinement: Parameter Synthesis for Probability above 0.15. A $*$ next to a value means that it was estimated by extrapolating from the run-times for problems with a smaller number of hyperboxes. For example, the 13.9 days estimate for $262,144$ hyperboxes was obtained by extrapolating from the 9.8 hours it took for $4,096$ hyperboxes.

tive search columns in Table 5.1 provide a lower bound for the time taken by Algorithm 4. Here, we just studied the performance of guided refinement on various abstractions with 64, 4096 and 262144 hyperboxes. In each of these cases, a naive algorithm would need to analyze all of these hyperboxes by refining them into even smaller hyperboxes. We found that an analysis of the 64 hyperboxes formed at the first step showed that only 10 of them needed to be refined and further analyzed. While the analysis of 4096 boxes takes 9.8 hours, the analysis of the 10 well-chosen boxes (by exploring 640 hyperboxes) takes about one and a half hours. The savings become much more impressive as we refine the size of the hyperbox and increase the number of hyperboxes. If we were to analyze all of $262,144$ boxes, we *estimate* (via extrapolation) that it would take over 13.9 days. On the other hand, if we use abstraction refinement based on the monotonicity argument, we only need to analyze at most $15,744$ hyperboxes and that takes about 20 hours. We further estimate that the exhaustive analysis of all $16,777,216$ smaller hyperboxes would take about 1.7 years, but that the refinement algorithm would take no more than 38 days.

When abstractions can be built over a space of models (as opposed to over parameters), abstraction refinement is often an efficient technique to employ. This *may* lead to considerable savings in many cases. On the other hand, one can cleverly construct cases where abstraction refinement may have to work as hard as the original analysis algorithm. However, this is not possible in our case, because we are performing abstractions over the parameters. The monotonicity of the parameter space ensures that abstraction refinement will always run faster in synthesizing kinetic parameters of stochastic biochemical systems.

5.6.3 Parameter Estimation using Gradient Descent (Algorithm 6)

Gradient descent based algorithms for synthesizing single parameter values are truly scalable to high dimensions as they do not need to do a search (exhaustive or otherwise) of the entire probability space. We applied our gradient descent algorithm to the problem of synthesizing the lb and bb kinetic parameters for the Cell Cycle model. We wanted to find a parameter value that satisfies the following Probabilistic Bounded Metric Temporal Logic specification:

$$Pr_{\geqslant 0.45} \ [\, \textbf{True}\ \mathbf{U}^{[60,60]} cyclin\_bound = 0]$$

The results of our algorithm are plotted in Figure 5.9. The algorithm suggests the parameter tuple $(0.349, 0.124)$ as a parameter value that satisfies the specification. The algorithm took only 21 minutes to produce this result. We further ran our algorithm to find the maximal value in the parameter space between 0.01 and 200 for both the parameters.

Our algorithm reported that the maximum value of the probability is 0.539 and lies at the point $(0.0101, 0.0101)$ which is the closest point we sampled to one of the corners of the given parameter space.

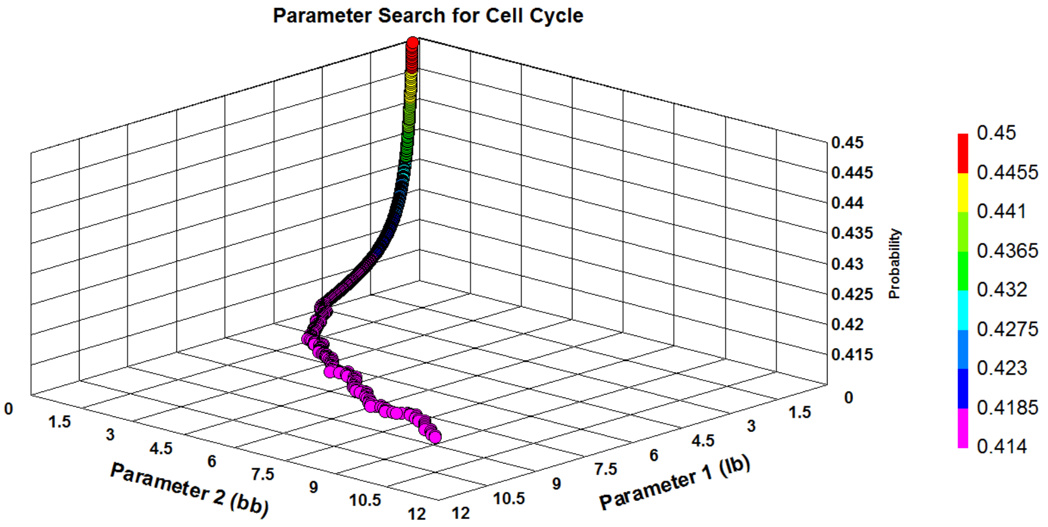


Figure 5.9: Parameter Search in action for the Cell Cycle Model.

Our cell cycle model has 11 kinetic parameters that can be varied. We considered the parameter synthesis problem involving all of the eleven parameters with the aim of satisfying the following formula:

$$Pr_{\leqslant 0.4}\ [\ \mathbf{True}\ \mathbf{U}^{[60,60]} cyclin\_bound = 0]$$

We restricted search space for the kinetic parameters between 0.001 and 1. Our binary search based implementation of Algorithm 6 took 3.4 hours to report the parameter value $(0.112512, 0.5005, 0.5005, 0.5005, 0.5005, 0.5005, 0.5005, 0.165856, 0.5005, 0.5005, 0.5005)$

that satisfies the formula with probability 0.3986.

We then searched for a parameter that satisfies the following formula:

$$Pr_{\geqslant 0.99} \ [\ \mathbf{True} \ \mathbf{U}^{[60,60]} cyclin\_bound = 0]$$

Our implementation took 52.3 minutes to produce the following parameter values that satisfy the property with probability exceeding 0.9902: $(0.5005, 0.5005, 0.5005, 0.5005, 0.5005, 0.326276, 0.5005, \ 0.5005, 0.5005, \ 0.5005, \ 0.5005)$. We also asked our algorithm to find the maximum value of the parameter possible and we obtained 0.9999 as the answer within the probability space we were searching.

Chapter 6

Conclusion and Future Work

In this thesis, we have presented a new Statistical Model Checking algorithm based on a sequential version of the Bayes Factor test. Our Bayesian algorithm directly solves the composite hypothesis testing problem posed by Statistical Model Checking and does not need to reduce it to a simple hypothesis testing problem. We have also complemented our algorithm with the analysis of its frequentist properties including the Type I/II error probabilities and a proof of termination of the algorithm.

We have also suggested a cost based approach to Statistical Model Checking. This is useful in those practical settings where a lower bound on the cost of generating each sample and an upper bound on the cost of making an incorrect decision are both known. We also study the influence of the cost of making an incorrect decision on the number of samples needed by our algorithm.

While traditional Statistical Model Checking uses independent and identically distributed samples from a system, such an approach is not very useful for detecting rare errors in systems. We have also suggested a new Bayesian Statistical Model Checking algorithm that allows one to use non-identically distributed samples. Our results suggest that a long-term neutrality of the sampling procedure is sufficient to ensure both the termination of the algorithm and maintain statistical guarantees on the Type I/II errors for our algorithm.

We have also investigated the synthesis of stochastic biochemical models against behavioral properties. Kinetic parameters of biochemical models are often hard to measure experimentally. On the other hand, a number of experimentally known facts about the model are documented in literature. We have proposed a new framework for the synthesis of kinetic parameters from known facts about biochemical models. In the process, we have characterized the probability of a biochemical model satisfying a specification as a uniformly continuous function of the kinetic parameters. Our proof uses the fact that the samples used by survey sampling based statistical estimation methods are independent of the value of the kinetic parameters of the biochemical model. We believe that this is the first time that survey sampling has been used as a proof mechanism to prove correctness of an algorithm in biochemical model simulation or discovery.

6.1 Conclusions

In this thesis, we have presented a number of algorithms for the validation and synthesis of various subclasses of stochastic systems. In Table 6.1, we study the various algorithms that we have discussed in the thesis including expectations from the stochastic model that can be analyzed by these algorithms.

6.1.1 Bayesian Statistical Model Checking

Our study of the Bayesian Statistical Model Checking algorithm indicates i.i.d. sampling is a good choice when analyzing *soft* stochastic systems, like biochemical models, where the probability of an adapted finitely monitorable specification being true is neither very close to one nor zero. In scenarios where a stochastic system may satisfy a formula with probability very close to unity or zero, i.i.d. sampling methods are not efficient at discovering these rare behaviors or so-called “black swans[1]" [Tal07]. In these cases, non-i.i.d. sampling strategies (like Algorithm 3) are preferred. Alternatively, symbolic testing approaches combined with statistical methods[AKRS08] may be a more attractive framework for such hard verification problems. However, we note that many stochastic systems of interest, like those arising from biochemical systems, are often too large for analysis by symbolic methods. Fortunately, many of these systems can be simulated, numerically, and so simulation-based verification techniques, like ours, can still be used when symbolic methods fail.

Our Bayesian approach provides a natural framework for incorporating information about prior beliefs about the system being analyzed and for analyzing models with probability distributions over parameters. This situation arises frequently in modeling of biochemical systems where the model is often parametric and we only know probability distributions on these model parameters. The ability to explicitly incorporate prior beliefs can

[1] The hypothesis that all swans are white had an incredibly strong statistical support based on millions of samples until the discovery of black swans in 1697 by European explorers in Western Australia.

Algorithm	Nature of Stochastic Model	Purpose of Algorithm	Additional Remarks
Bayesian Statistical Model Checking	Discrete Time Markov Chains, Continuous Time Markov Chains, (Discretely Sampled) Stoch. Differential Eqns	To validate if a stochastic model satisfies a given behavioral specification with at least a given probability	Bayes Factor as the criterion for termination
Cost based Bayesian Statistical Model Checking	Discrete Time Markov Chains, Continuous Time Markov Chains, (Discretely Sampled) Stoch. Differential Eqns	To validate if a stochastic model satisfies a given behavioral specification with at least a given probability	Cost of simulation and cost of making an error as the criterion for termination
Non-i.i.d. Bayesian Statistical Model Checking	Discrete Time Markov Chains specified as Probabilistic Reactive Modules, (Discretely Sampled) Stoch. Differential Eqns	To validate if a stochastic model satisfies a given behavioral specification with at least a given probability	Bayes Factor as the criterion for termination
Uniform Continuity based Parameter Synthesis	Biochemical Continuous Time Markov Chains	To synthesis chemical kinetics parameters such that the model satisfies a given probabilistic specification	An acceptable error tolerance of the boundary of the synthesized parameter space.
Abstraction Refinement based Parameter Synthesis	Monotonic Parameter Spaces in Biochemical Continuous Time Markov Chains	To synthesis chemical kinetics parameters such that the model satisfies a given probabilistic specification	An acceptable error tolerance of the boundary of the synthesized parameter space.
Parameter Search	Monotonic Parameter Spaces in Biochemical Continuous Time Markov Chains	To find a parameter values that maximizes the probabiity of a given specification being true	An acceptable error tolerance of the boundary of the synthesized parameter space.

Table 6.1: Summary of Algorithms

also accelerate the model validation process. On the other hand, the computational cost of the Bayesian approach may not make it amenable for deployment in applications where energy or computational resource may be a constraint, like embedded systems. If a Statistical Model Checking algorithm needs to be incorporated into a smart embedded system, we recommend the use of Younes' technique, which is computationally attractive [You04].

6.1.2 Discovery of Stochastic Biochemical Systems against Behavioral Specifications

We have introduced new algorithms to discover kinetic parameter values that enable a biochemical model to satisfy a Probabilistic Adapted Finitely Monitorable logic specification. The specification captures the biological knowledge that is known about the biochemical system being modeled. We applied our algorithms to two benchmark models from the literature, viz. Fibroblast Growth Factor and Cell Cycle model. We demonstrated that our abstraction-refinement based algorithm is capable of synthesizing six parameters simultaneously. To the best of our knowledge, this is the largest number of parameters that have been synthesized at once for stochastic models against a given formula. Moreover, our gradient search algorithm is capable of finding the single parameter combination that maximizes the probability of the formula being true over 11 parameters simultaneously.

A key feature of theses algorithms is that they can also demonstrate the *infeasibility* of a model with respect to a high level behavioral specification in a given parameter space. This is a very useful debugging tool for biochemical modeling, and ensures that the modeler does not waste her time searching for a non-existent parameter combinations. The gradient search algorithm is capable of carrying out infeasibility analysis in very high dimensional parameter spaces.

An important feature of our synthesis algorithm based on uniform continuity is the lack of any assumptions about the shape of the parameter space that we seek to synthesize. This is a crucial different between our algorithm and approaches based on learning Bayesian probability distributions over the parameter space. In the Bayesian learning approach [GCSR03], a prior distribution over the probability with which a parameter value satisfies a given specification is updated. Then, sampling approaches are used to update this prior information by actually computing the probability of the parameter value satisying a specification for some parameter values. A serious weekness in the approach lies in its inability to change the effect of the form of the chose prior distribution on the final outcome. For example, if one tries to fit a Gaussian distribution to the parameter space, one has already made the assumption that the subset of the parameter space that satisfies the given specification with the required probability is connected and symmetric. Similar constraints hold for other choices of probability distributions. Such knowledge about the parameter space being synthesized is not usually available before we start the synthesis procedure and hence, the choice of a suitable probabilit distribution becomes difficult.

6.2 Future Work

Several interesting directions of future work remain open. We have used an off-the-shelf simulator for biochemical systems (BioNetGen [FBGH05, FBH08, FBH05, BFGH04]) that isn't designed to draw samples from biochemical models with different kinetic parameters. It is possible to design a more efficient simulator that can take account of variation in kinetic parameters and draw samples from a family of parameterized biochemical systems more efficiently. The intuitive idea is that the same pair of random numbers can be used to generate transitions in all models of a parameterized family. Also, many paths in closely related parameterized models would share a lot of common transitions.

Our experiments have used rather restricted PAFM specifications like Probabilistic Bounded Linear Temporal Logic (PBLTL) and Continuous Specification Logic (CSL). One can envision the development of a specialized logic for simulation based validation of biochemical stochastic systems. VLSI design and validation engineers use a syntactic sugar for Linear Temporal Logic called ForSpec [AFF$^+$02] (developed at Intel). It would be useful to collect specifications from biochemical modelers, and develop a syntactic sugar specially meant for biochemical systems.

Bayesian Statistical Model Checking algorithm can be extended to accommodate nested probability operators. While nested probability operators are permitted in Continuous Specification Logic (CSL), they are not used very often in practice. However, it is still a theoretical curiosity to extend the algorithm to analyze properties with nested probability operators.

The most intriguing problem is to perform statistical validation of non-deterministic systems with no natural notion of a probability measure, such as hardware and software systems. More research is needed to determine if any of the methods presented in this thesis can be adapted to such systems.

We have shown how non-i.i.d. sampling can be used with Stochastic differential equations and probabilistic reactive modules. An interesting direction of future work is to extend the non-i.i.d. sampling based Bayesian statistical model checking algorithm to Continuous Time Markov Chains. The probability of a transition in a CTMC is a function of the transition rates: $P(s \to s') = k_{s \to s'} e^{-\sum_t k_{s \to t}}$ where s, s' and t are state of the CTMC and $k_{s \to s'}$ denotes the rate of transition between s and s'. Clearly, preserving the geometric average of the transition rates does not preserve the geometric average of the transition probabilities.

Similarly, it is not clear how one would develop *fair* sampling strategies for non-i.i.d. sampling from complex probabilistic distributions on inputs of otherwise deterministic models. Intuitively, we need to preserve the geometric average of the probability distributions on each input. However, the exact nature of the geometric average would depend upon the form of the probability distribution. We are investigating the forms of probability distributions that give rise to a tractable approach for developing fair non-i.i.d. sampling approaches.

Our results and constructive proofs for discovery of kinetic parameters in biochemical stochastic systems can be extended to study the impact of variation in different parameters on the probability of an adapted finitely monitorable formula being true on a model. Such an algorithmic sensitivity analysis of the probability of a formula being true with respect to the various parameters can then be used as a *preprocessing step* to guide parameter search algorithms in very high dimensional parameter spaces. For example, consider Figure 6.1 which shows the variation in probability as we vary the lb and bb parameters of the cell cycle model. It is clear that variation along one of the parameters impacts the model much more than along the other dimension. Such a preliminary sensitivity analysis could be used to pick a small number of parameters out of a large number of unknown parameters before parameter synthesis is attempted on the smaller parameter set. We also note that the proof of Lemma 1 provides a framework of computing sensitivity using statistical sampling and can be developed into an efficient sensitivity analysis algorithm for stochastic biochemical models.

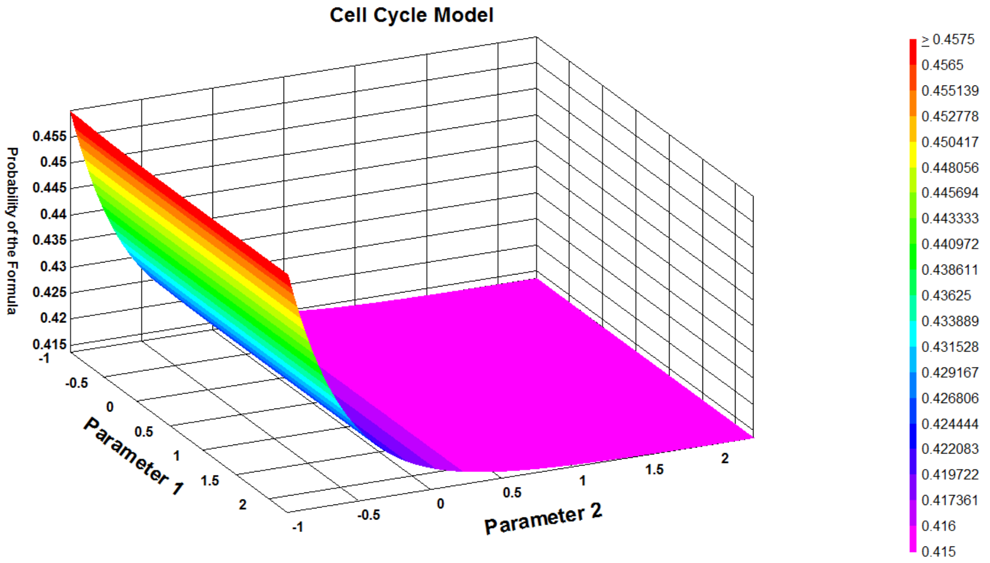


Figure 6.1: Probability of formula [$\textbf{True}\ \mathbf{U}^{60} cyclin\_bound = 0$] for Cell Cycle Model. Note that the variation along parameter 1(lb) is much smaller than variation along parameter 2(bb).

Another interesting direction for future work is to develop a monitoring framework for stochastic biochemical systems. It is unrealistic to assume that biochemists will ever translate their knowledge into fragments of temporal logics. On the other hand, MATLAB and even high level languages like C are now standard in many undergraduate programs across the world. It is important to construct a suitable framework for developing monitors in these languages that can be used directly by biochemists. While some work has been done on automatically mining formal knowledge bases or ontologies from biochemical literature, the use of these ontologies to validate models is limited at best. It would be interesting to bridge the gap and provide formal methods based tools to verify models against existing ontologies.

We also note that existing stochastic biochemical models are often hard-wired with a “best-guess" value of the parameters that makes the model “work". Unfortunately, when different models are combined in a modular fashion (e.g., combining models of different pathways), those models that “work" in isolation might not work together as components of the larger system. These inconsistencies will come into focus as the science of Computational Systems Biology matures, and we begin putting models built by different scientists together. An interesting question that we are investigating is the *re-synthesis* of parameter regimes when putting models together as components. In this context, it is clear that biochemical models should not be hard-wired with “best-guess" values of kinetic parameters. Instead, a model should be accompanied with a set of kinetic parameters that enable the model to satisfy every known behavior about the model. When models are put together, we will have new properties and biological insights about the larger and more complex biochemical system. This will enable us to refine our models and parameter values.

An important contribution of the thesis is a new technique for constructing proofs about stochastic biochemical models based on survey sampling. Parameterized biochemical models give rise to families of CTMCs, each of which has a different probability measure over the set of possible paths. The key point, however, is that the set of possible paths is the same. Further, the sample paths that (do not) satisfy the property also remain unchanged across all CTMCs in a family of CTMCs. Thus, survey sampling provides a natural framework to develop analytic arguments about biochemical models. We believe that this technique for developing proofs may be important in areas beyond model validation and parameter synthesis. Particularly, the formal sensitivity analysis of τ-leaping methods [CGP07, AGK09] may be susceptible to an analysis along these lines. We are currently investigating this direction along with our collaborators.

The synthesis algorithm we discussed assumes that the initial state of the stochastic biochemical model is completely specified. One is often interested in understanding the influence of initial configuration of the stochastic biochemical model on the parameters synthesized using behavioral specifications. One possibility is to use a probability distribution on the initial configurations of the stochastic model and then use statistical sampling approaches to compute Bayesian averages of the probability of the specification being true over all possible initial configurations.

Our algorithms based on abstraction refinement require that the probability of a specification being true on a model be monotonic in the parameter space. It may be possible to develop a hierarchy of abstractions using upper bounds on the partial derivatives of the probability of the specification being true with respect to each of the parameters. The central idea would be to bound the probability of the specification being true in a parameter space by sampling one or more points in the parameter space and then reasoning that the probability values can not change any faster than that dictated by the upper bounds on the partial derivative of the probability with respect to reach of the parameters.

While the algorithms for discovery of stochastic biochemical models have focused on CTMCs, DTMCs naturally capture the semantics of Agent Based Models (ABMs).

We have developed discovery algorithms based on uniform continuity and monotonicity arguments. Because of the discrete nature of DTMCs and the finite number of finite-length paths, the proofs for the correctness of discovery algorithms for DTMCs are simpler [JKL10].

Another interesting direction of research that we are currently pursuing is the discovery of parameters in stochastic differential equation (SDE) models. Financial models are often modeled as continuous time SDEs and the synthesis of such models from behavioral specifications would be of interest. This is particularly challenging because even the number of possible states of the solution to a SDE is infinite. Hence, it is difficult to adapt our proofs directly to the setting of stochastic differential equations. There has been recent work on developing robustness arguments for solutions of ordinary differential equations and applying them for model discovery [DCL10]. However, it is known that Brownian Motion is unbounded everywhere except at zero and thus, reachability is trivially true. Hence, adapting the robustness argument for reachability in ODEs [DCL10] is not feasible for SDEs.

Appendix A

Proofs

A.1 Termination of Bayesian Model validation Algorithm

Lemma 3 *If ρ and $A = [a_0, a_1]$ are strongly $\delta-$separated, and $b_0 = -\log \delta$, then, for all probability v on A, for all n,*

$$\mathit{Aff}\left(f(x_1|\rho) \dots f(x_n|\rho), \int_0^1 f(x_1|u) \dots f(x_n|u) v(u) du \right) < e^{-nb_0} \tag{A.1}$$

Proof 24 *Proof by Induction:*

Base Case: $\mathit{Aff}\Big(f(x_1|\rho), \int_0^1 f(x_1|u) v(u) du \Big) < \delta$

Inductive Hypothesis: $\mathit{Aff}\Big(f(x_1|\rho) \dots f(x_n|\rho), \int_0^1 f(x_1|u) \dots f(x_n|u) v(u) du \Big) < e^{-nb_0}$

Inductive Step:

$$\begin{aligned}
& \mathit{Aff}\left(f(x_1|\rho)\ldots f(x_n|\rho)f(x_{n+1}|\rho), \int_0^1 f(x_1|u)\ldots f(x_n|u)f(x_{n+1}|u)v(u)du\right) \\
= \ & \int_0^1 \sqrt{f(x_1|\rho)\ldots f(x_n|\rho)f(x_{n+1}|\rho)\int_\Theta f(x_1|u)\ldots f(x_n|u)f(x_{n+1}|u)dv}\ \ d\mu \\
\leqslant \ & \int_0^1 \sqrt{f(x_1|\rho)\ldots f(x_{n+1}|\rho)(\int_\Theta [f(x_1|u)\ldots f(x_n|u)]^2 dv)^{1/2}\ \ (\int_\Theta [f(x_{n+1}|u)]^2 dv)^{1/2}}\ \ d\mu \\
\leqslant \ & \int_0^1 \sqrt{f(x_1|\rho)\ldots f(x_{n+1}|\rho)(\int_\Theta f(x_1|u)\ldots f(x_n|u)dv)\ \ (\int_\Theta f(x_{n+1}|u)dv)}\ \ d\mu \\
\leqslant \ & \int_0^1 \sqrt{f(x_1|\rho)\ldots f(x_n|\rho)(\int_\Theta f(x_1|u)\ldots f(x_n|u)dv)\ \ f(x_{n+1})(\int_\Theta f(x_{n+1}|u)dv)}\ \ d\mu \\
\leqslant \ & \sqrt{\int_0^1 f(x_1|\rho)\ldots f(x_n|\rho)\int_\Theta f(x_1|u)\ldots f(x_n|u)dv\ \ d\mu}\ \sqrt{\int_0^1 f(x_{n+1})\int_\Theta f(x_{n+1}|u)dv\ \ d\mu} \\
\leqslant \ & \int_0^1 \sqrt{f(x_1|\rho)\ldots f(x_n|\rho)\int_\Theta f(x_1|u)\ldots f(x_n|u)dv}\ \ d\mu\ \ \int_0^1 \sqrt{f(x_{n+1})\int_\Theta f(x_{n+1}|u)dv}\ \ d\mu \\
< \ & e^{-nb}.e^{-b} \qquad\qquad = e^{-(n+1)b}
\end{aligned}$$

The above steps make repeated use of Holder's inequatilty for integrals and expectations. Hence, proved by induction.

Theorem 23 (Bayesian Consistency Theorem [CR08]) *If x_i are* i.i.d. *samples of the Bernoulli random variable $X_i (1 \leqslant i \leqslant n)$ with probability of success ρ such that ρ lies is in the KL support of the prior g, $A = [a_0, a_1]$ is strongly δ- separated from ρ (for some $\delta > 0$), and the prior probability measure on A is finite, then the posterior probability of A decreases exponentially to 0 almost everywhere.*

$$P\left(\frac{\displaystyle\int_{a_0}^{a_1} f(x_1|u)\cdots f(x_n|u)\cdot g(u)\;du}{\displaystyle\int_{0}^{1} f(x_1|u)\cdots f(x_n|u)\cdot g(u)\;du} \geqslant e^{-nb}\; i.o.\right) = 0$$

Here, b is a constant and the abbreviation $i.o.$ stands for infinitely often.

Proof 25 *For the details of the proof, please see [CR08].*

$$P\left(\sqrt{\int_{a_0}^{a_1} \frac{f(x_1|u)}{f(x_1|\rho_0)}\cdots\frac{f(x_n|u)}{f(x_n|\rho_0)}\cdot g(u)\;du} \geqslant e^{-nb}\right)$$

$$\leqslant e^{nb}\;E\left[\sqrt{\int_{a_0}^{a_1} \frac{f(x_1|u)}{f(x_1|\rho_0)}\cdots\frac{f(x_n|u)}{f(x_n|\rho_0)}\cdot g(u)\;du}\right] \qquad \ldots\textit{Markov's Inequality}$$

$$= e^{nb}\sum_{x\in X^n}\sqrt{\int_{a_0}^{a_1} \frac{f(x_1|u)}{f(x_1|\rho_0)}\cdots\frac{f(x_n|u)}{f(x_n|\rho_0)}\cdot g(u)\;du}\;\; f(x_1|\rho_0)\ldots f(x_n|\rho_0) \qquad \ldots\textit{Definition of Expectation}$$

$$= e^{nb}\sum_{x\in X^n}\sqrt{\int_{a_0}^{a_1} f(x_1|u)\cdots f(x_n|u)\cdot g(u)\;du}\;\;\sqrt{f(x_1|\rho_0)\ldots f(x_n|\rho_0)} \qquad \ldots\textit{Algebraic Manipulation}$$

$$= e^{nb}\sum_{x\in X^n}\sqrt{\frac{\displaystyle\int_{a_0}^{a_1} \frac{f(x_1|v)}{f(x_1|\rho_0)}\cdots\frac{f(x_n|v)}{f(x_n|\rho_0)}\cdot g(v)\;dv}{\displaystyle\int_{0}^{1} \frac{f(x_1|v)}{f(x_1|\rho_0)}\cdots\frac{f(x_n|v)}{f(x_n|\rho_0)}\cdot g(v)\;dv}}$$

$$\sqrt{\int_{a_0}^{a_1} f(x_1|u)\cdots f(x_n|u)\cdot\frac{\displaystyle\int_{0}^{1} \frac{f(x_1|v)}{f(x_1|\rho_0)}\cdots\frac{f(x_n|v)}{f(x_n|\rho_0)}\cdot g(v)\;dv}{\displaystyle\int_{a_0}^{a_1} \frac{f(x_1|v)}{f(x_1|\rho_0)}\cdots\frac{f(x_n|v)}{f(x_n|\rho_0)}\cdot g(v)\;dv} g(u)\;du}\;\;\sqrt{f(x_1|\rho_0)\ldots f(x_n|\rho_0)} \qquad \ldots\textit{Algebraic Manipulation}$$

$$= e^{nb}\sum_{x\in X^n}\sqrt{\frac{\displaystyle\int_{a_0}^{a_1} \frac{f(x_1|v)}{f(x_1|\rho_0)}\cdots\frac{f(x_n|v)}{f(x_n|\rho_0)}\cdot g(v)\;dv}{\displaystyle\int_{0}^{1} \frac{f(x_1|v)}{f(x_1|\rho_0)}\cdots\frac{f(x_n|v)}{f(x_n|\rho_0)}\cdot g(v)\;dv}}$$

$$\sqrt{\int_{a_0}^{a_1} f(x_1|u)\cdots f(x_n|u)\cdot g^*(u)\;du}\;\;\sqrt{f(x_1|\rho_0)\ldots f(x_n|\rho_0)} \qquad \ldots g^*\textit{ is restriction of } g \textit{ on } [a_0,a_1]$$

$$\leqslant e^{nb}\max_{x\in X}\sqrt{\frac{\displaystyle\int_{a_0}^{a_1} \frac{f(x_1|v)}{f(x_1|\rho_0)}\cdots\frac{f(x_n|v)}{f(x_n|\rho_0)}\cdot g(v)\;dv}{\displaystyle\int_{0}^{1} \frac{f(x_1|v)}{f(x_1|\rho_0)}\cdots\frac{f(x_n|v)}{f(x_n|\rho_0)}\cdot g(v)\;dv}}$$

$$\sum_{x\in X^n}\sqrt{\int_{a_0}^{a_1} f(x_1|u)\cdots f(x_n|u)\cdot g^*(u)\;du}\;\;\sqrt{f(x_1|\rho_0)\ldots f(x_n|\rho_0)} \qquad \ldots\textit{Algebraic Manipulation}$$

$$\leqslant e^{nb} \max_{x \in X^n} \sqrt{\frac{\displaystyle\int_{a_0}^{a_1} \frac{f(x_1|u)}{f(x_1|\rho_0)} \cdots \frac{f(x_n|u)}{f(x_n|\rho_0)} \cdot g(u)\ du}{\displaystyle\int_0^1 \frac{f(x_1|u)}{f(x_1|\rho_0)} \cdots \frac{f(x_n|u)}{f(x_n|\rho_0)} \cdot g(u)\ du}}$$

$$\mathit{Aff}\left(f(x_1|\rho_0) \ldots f(x_n|\rho_0), \int_{a_0}^{a_1} f(x_1|u) \ldots f(x_n|u) g^*(u) du \right) \quad \ldots \textit{Definition of Affinity}$$

$$\leqslant e^{nb}\ \sqrt{\Pi_{max}([a_0, a_1])}\ \ e^{-nb_0} \quad \ldots \textit{Using} \quad \textit{Eqn. 4.5}$$

$$\leqslant e^{nb}\ (1)\ \ e^{-nb_0} \quad \ldots \textit{Probability is at most 1}$$

Choosing $b = b_0/4$, we see that

$$P(\sqrt{\int_{a_0}^{a_1} \frac{f(x_1|u)}{f(x_1|\rho_0)} \cdots \frac{f(x_n|u)}{f(x_n|\rho_0)} \cdot g(u)\ du} \geqslant e^{-nb} \text{ i.o.}) = 0$$

We know that $\lim_{n\to\infty} e^{nb}\sqrt{\displaystyle\int_0^1 \frac{f(x_1|u)}{f(x_1|\rho_0)} \cdots \frac{f(x_n|u)}{f(x_n|\rho_0)} \cdot g(u)\ du} = \infty$ a.s. Hence, the result follows.

A.2 Proof of Uniform Continuity

Theorem 24 (Uniform Continuity of Logarithm of Path Probability Density) *For every $\epsilon \in \mathcal{R}^+$, there exists $\delta \in \mathcal{R}^+$ such that $|\log P'(\sigma) - \log P(\sigma)| \leqslant \epsilon$ holds whenever $|\log {k_j}' - \log k_j| \leqslant \delta$, for all j $(1 \leqslant j \leqslant n)$.*

Proof 26 *We know that the probability density of moving from state s_i to state s_{i+1} by* *executing reaction r_{j_i} after time Δ_i is given by*

$$P(s_i \xrightarrow{\Delta_i} s_{i+1}) = k_{j_i} x_1(s_i)^{\alpha_1^{j_i}} \ldots x_m(s_i)^{\alpha_m^{j_i}} \exp\left(-\sum_{h=1}^{n} k_h x_1(s_i)^{\alpha_1^h} \ldots x_m(s_i)^{\alpha_m^h} \Delta_i\right)$$

Taking logarithms on both sides,

$$\begin{aligned}\log P(s_i \xrightarrow{\Delta_i} s_{i+1}) &= \log\left(k_{j_i} x_1(s_i)^{\alpha_1^{j_i}} \ldots x_m(s_i)^{\alpha_m^{j_i}}\right) - \sum_{h=1}^{n} k_h x_1(s_i)^{\alpha_1^h} \ldots x_m(s_i)^{\alpha_m^h} \Delta_i \\ &= \log\left(k_{j_i} \gamma_{(i,i+1)}^{j_i}\right) - \sum_{h=1}^{n} k_h \gamma_{(i,i+1)}^{h} \Delta_i \\ &= \log k_{j_i} + \log\left(\gamma_{(i,i+1)}^{j_i}\right) - \sum_{h=1}^{n} k_h \gamma_{(i,i+1)}^{h} \Delta_i\end{aligned}$$

Here, $\gamma_{(i,i+1)}^{h} \overset{def}{\equiv} x_1(s_i)^{\alpha_1^h} \ldots x_m(s_i)^{\alpha_m^h}$ *is a quantity independent of* k_h $(1 \leqslant h \leqslant n)$. *And so,* $|\log P(s_i \xrightarrow{\Delta_i} s_{i+1}) - \log P'(s_i \xrightarrow{\Delta_i} s_{i+1})|$

$$\begin{aligned} &= |\,[\log k_{j_i} + \log\left(\gamma_{(i,i+1)}^{j_i}\right) - \sum_{h=1}^{n} k_h \gamma_{(i,i+1)}^{h} \Delta_i] \\ &\qquad - [\log k'_{j_i} + \log\left(\gamma_{(i,i+1)}^{j_i}\right) - \sum_{h=1}^{n} k_h{}' \gamma_{(i,i+1)}^{h} \Delta_i]\,| \\ &= |\,[\log k_{j_i} - \sum_{h=1}^{n} k_h \gamma_{(i,i+1)}^{h} \Delta_i] - [\log k'_{j_i} - \sum_{h=1}^{n} k_h{}' \gamma_{(i,i+1)}^{h} \Delta_i]\,| \\ &= |\,\log k_{j_i} - \log k'_{j_i} + \sum_{h=1}^{n} (k'_h - k_h) \gamma_{(i,i+1)}^{h} \Delta_i\,| \\ &\leqslant |\log k_{j_i} - \log k'_{j_i}| + |\sum_{h=1}^{n} (k'_h - k_h) \gamma_{(i,i+1)}^{h} \Delta_i| \qquad \textit{(Triangle Inequality)}\end{aligned}$$

$$
\begin{aligned}
&\leqslant \; |\log k_{j_i} - \log k'_{j_i}| + \gamma^{max}_{(i,i+1)}\Delta_i|\sum_{h=1}^{n}(k'_h - k_h)| \qquad (\gamma^{max}_{(i,i+1)} \overset{def}{\equiv} \max_{1\leqslant h\leqslant n} \gamma^{h}_{(i,i+1)}) \\
&\leqslant \; |\log k_{j_i} - \log k'_{j_i}| + \gamma^{max}_{(i,i+1)}\Delta_i \sum_{h=1}^{n}|k'_h - k_h| \qquad \textit{(Triangle Inequality)}
\end{aligned}
$$

Consider the finite path $\sigma \equiv s_0 \xrightarrow{\Delta_0} s_1 \cdots \xrightarrow{\Delta_{l-1}} s_l$*. Let* $P(\sigma)$ *be the probability density associated with the path in the model* $\mathcal{M}(\Theta)$ *and* $P'(\sigma)$ *be the probability density associated with the path in the model* $\mathcal{M}(\Theta')$*.*

$$
\begin{aligned}
P(\sigma) &= P(s_0 \xrightarrow{\Delta_0} s_1) \times \cdots \times P(s_{l-1} \xrightarrow{\Delta_{l-1}} s_l) \\
\Rightarrow \log P(\sigma) &= \log P(s_0 \xrightarrow{\Delta_0} s_1) + \cdots + \log P(s_{l-1} \xrightarrow{\Delta_{l-1}} s_l)
\end{aligned}
$$

So,

$$
\begin{aligned}
& |\; \log P(\sigma) - \log P'(\sigma) \;| \\
\leqslant \;& \sum_{i=0}^{l-1} |\log k_{j_i} - \log k'_{j_i}| + \sum_{i=0}^{l-1}\Big(\gamma^{max}_{(i,i+1)}\Delta_i \sum_{h=1}^{n}|k'_h - k_h|\Big) \\
\leqslant \;& l \max_{j_i, i\in[0,l-1]} |\log k_{j_i} - \log k'_{j_i}| + \Big(\sum_{h=1}^{n}|k'_h - k_h|\Big)\sum_{i=0}^{l-1}\Big(\gamma^{max}_{(i,i+1)}\Delta_i\Big) \\
\leqslant \;& l|\log k_j - \log k'_j|^{max} + \gamma^{max}\Big(\sum_{h=1}^{n}|k'_h - k_h|\Big)\sum_{i=0}^{l-1}\Delta_i
\end{aligned}
$$

where $\gamma^{max} \overset{def}{\equiv} \max_i \gamma^{max}_{(i,i+1)}$ *and* $|\log k_j - \log k'_j|^{max} \overset{def}{\equiv} \max_{j_i, i\in[0,l-1]} |\log k_{j_i} - \log k'_{j_i}|$*.*

And so,

$$\begin{aligned}
& \mid \log P(\sigma) - \log P'(\sigma) \mid \\
\leqslant\ & l|\log k_j - \log k'_j|^{max} + \gamma^{max}\Big(\sum_{h=1}^{n} |k'_h - k_h|\Big)\Delta_{total}, \textit{ where } \Delta_{total} \equiv \sum_{i=0}^{l-1} \Delta_i \\
\leqslant\ & l|\log k_j - \log k'_j|^{max} + \gamma^{max}\Big(\sum_{h=1}^{n} M\left(e^{|\log k'_h - \log k_h|} - 1\right)\Big)\Delta_{total}
\end{aligned}$$

To show that $\mid \log P(\sigma) - \log P'(\sigma) \mid \leqslant \epsilon$*, it is sufficient to show that the following holds:*

$$l|\log k_j - \log k'_j|^{max} + \gamma^{max}\Big(\sum_{h=1}^{n} M\left(e^{|\log k'_h - \log k_h|} - 1\right)\Big)\Delta_{total} \leqslant \epsilon$$

.

From the statement of our theorem, we know that $|\log k_j - \log k'_j|^{max} \leqslant \delta$*. One can verify that the following choice of* δ *is sufficient to show that* $\mid \log P(\sigma) - \log P'(\sigma) \mid \leqslant \epsilon$*:*

$$\begin{aligned}
\delta &= \min\left(\frac{\epsilon}{l(n+1)}, \log\left(\frac{\epsilon}{(n+1)\max(\gamma^{max} M \Delta_{total}, 1)} + 1\right)\right) \\
&\overset{def}{\equiv} \delta(\epsilon, \mathcal{M}).
\end{aligned}$$

Bibliography

[AAB00] Aurore Annichini, Eugene Asarin, and Ahmed Bouajjani. Symbolic techniques for parametric reasoning about counter and clock systems. In E. Allen Emerson and A. Prasad Sistla, editors, *CAV*, volume 1855 of *Lecture Notes in Computer Science*, pages 419–434. Springer, 2000. 3.2

[ACH$^+$95] Rajeev Alur, Costas Courcoubetis, Nicolas Halbwachs, Thomas A. Henzinger, Pei-Hsin Ho, Xavier Nicollin, Alfredo Olivero, Joseph Sifakis, and Sergio Yovine. The algorithmic analysis of hybrid systems. *Theor. Comput. Sci.*, 138(1):3–34, 1995. 3.2

[ADG05] E. Asarin, T. Dang, and A. Girard. Hybridization methods for verification of non-linear systems. In *ECC-CDC'05 joint conference: Conference on Decision and Control CDC and European Control Conference ECC*, 2005. 3.2

[AFF$^+$02] Roy Armoni, Limor Fix, Alon Flaisher, Rob Gerth, Boris Ginsburg, Tomer Kanza, Avner Landver, Sela Mador-Haim, Eli Singerman, Andreas Tiemeyer, Moshe Y. Vardi, and Yael Zbar. The forspec temporal logic: A new temporal property-specification language. In *TACAS '02: Proceedings of the 8th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 296–211, London, UK, 2002. Springer-Verlag. 6.2

[AGK09] David F. Anderson, Arnab Ganguly, and Thomas G. Kurtz. Error analysis of tau-leap simulation methods, 2009. 6.2

[AKRS08] Rajeev Alur, Aditya Kanade, S. Ramesh, and K. C. Shashidhar. Symbolic analysis for improving simulation coverage of simulink/stateflow models. In *EMSOFT '08: Proceedings of the 8th ACM international conference on Embedded software*, pages 89–98, New York, NY, USA, 2008. ACM. 6.1.1

[Ams70] Arnold E. Amstutz. *Computer Simulation of Competitive Market Response*, volume 1 of *MIT Press Books*. The MIT Press, October 1970. 3

[AO91] K. R. Apt and E.-R. Olderog. *Verification of sequential and concurrent programs*. Springer-Verlag, 1991. 3

[BBB07] Alberto Bemporad, Antonio Bicchi, and Giorgio C. Buttazzo, editors. *Hybrid Systems: Computation and Control, 10th International Workshop, HSCC 2007, Pisa, Italy, April 3-5, 2007, Proceedings*, volume 4416 of *Lecture Notes in Computer Science*. Springer, 2007. A.2

[BCHG$^+$97] Christel Baier, Edmund M. Clarke, Vassili Hartonas-Garmhausen, Marta Z. Kwiatkowska, and Mark Ryan. Symbolic model checking for probabilistic processes. In Pierpaolo Degano, Roberto Gorrieri, and Alberto Marchetti-Spaccamela, editors, *ICALP*, volume 1256 of *Lecture Notes in Computer Science*, pages 430–440. Springer, 1997. 2.2, 2.3, 3.1, 4.5.4

[Ber85] J.O. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer-Verlag, 1985. 4

[Ber93] James O. Berger. *Statistical Decision Theory and Bayesian Analysis (Springer Series in Statistics)*. Springer, 1993. 4.4

[BFGH04] M. L. Blinov, J. R. Faeder, B. Goldstein, and W. S. Hlavacek. BioNetGen: software for rule-based modeling of signal transduction based on the interactions of molecular domains. *Bioinformatics*, 20(17):3289–3291, 2004. 6.2

[BG96] M. Broadie and P. Glasserman. Estimating security price derivatives using simulation. *Management Science*, 42:269–285, 1996. 3

[BHHK03] Christel Baier, Boudewijn R. Haverkort, Holger Hermanns, and Joost-Pieter Katoen. Model-checking algorithms for continuous-time markov chains. *IEEE Trans. Software Eng.*, 29(6):524–541, 2003. 2.2, 3.1

[BK08] Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking*. The MIT Press, May 2008. 1.1

[BM81] R. S. Boyer and J. S. Moore, editors. *The Correctness Problem in Computer Science*. Academic Press, 1981. 3

[BMMR01] Thomas Ball, Rupak Majumdar, Todd D. Millstein, and Sriram K. Rajamani. Automatic predicate abstraction of C programs. In *PLDI*, pages 203–213, 2001. 1.1

[BR02] T. Ball and S.K. Rajamani. The SLAM project: debugging system software via static analysis. In *POPL 2002: Proceedings of the 29th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 1–3, 2002. 1.1

[BS73] F. Black and M. Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81:637–654, 1973. 1.2, 4

[BYWB07] Gregory Batt, Boyan Yordanov, Ron Weiss, and Calin Belta. Robustness analysis and tuning of synthetic gene networks. *Bioinformatics*, 23(18):2415–2422, 2007. 3.2

[CB06] Frank Ciesinski and Christel Baier. Liquor: A tool for qualitative and quantitative linear time analysis of reactive systems. In *QEST*, pages 131–132. IEEE Computer Society, 2006. 3.1

[CCD+04] N. Chabrier-Rivier, M. Chiaverini, V. Danos, F. Fages, and V. Schãchter. Modeling and querying biomolecular interaction networks. *Theor. Comput. Sci.*, 325(1):25–44, 2004. 2.2

[CE82] Edmund M. Clarke and E. Allen Emerson. Design and synthesis of synchronization skeletons using branching-time temporal logic. In *Logic of Programs, Workshop*, pages 52–71, London, UK, 1982. Springer-Verlag. 2.3

[CF03] Nathalie Chabrier and François Fages. Symbolic model checking of biochemical networks. In Corrado Priami, editor, *CMSB*, volume 2602 of *Lecture Notes in Computer Science*, pages 149–162. Springer, 2003. 2.2, 3.2

[CFL+08] Edmund M. Clarke, James R. Faeder, Christopher James Langmead, Leonard A. Harris, Sumit Kumar Jha, and Axel Legay. Statistical model checking in biolab: Applications to the automated analysis of t-cell receptor signaling pathway. In Monika Heiner and Adelinde M. Uhrmacher, editors, *CMSB*, volume 5307 of *Lecture Notes in Computer Science*, pages 231–250. Springer, 2008. 2.2, 2.2, 3, 4.5

[CFS06a] Laurence Calzone, François Fages, and Sylvain Soliman. Biocham: an environment for modeling biological systems and formalizing experimental knowledge. *Bioinformatics*, 22(14):1805–1807, 2006. 3, 3.2

[CFS06b] Laurence Calzone, Francois Fages, and Sylvain Soliman. Biocham: an environment for modeling biological systems and formalizing experimental knowledge. *Bioinformatics*, 22(14):1805–1807, July 2006. 2.2

[CG04] F. Ciesinski and M. Größer. On probabilistic computation tree logic. In

Validation of Stochastic Systems, LNCS, 2925, pages 147–188. Springer, 2004. 3.1

[CGJ$^+$00] Edmund M. Clarke, Orna Grumberg, Somesh Jha, Yuan Lu, and Helmut Veith. Counterexample-guided abstraction refinement. In E. Allen Emerson and A. Prasad Sistla, editors, *CAV*, volume 1855 of *Lecture Notes in Computer Science*, pages 154–169. Springer, 2000. 5.4.1

[CGP99] E. M. Clarke, O. Grumberg, and D. A. Peled. *Model Checking*. MIT Press, Cambridge, MA, 1999. 2.3

[CGP07] Yang Cao, Daniel T. Gillespie, and Linda R. Petzold. Adaptive explicit-implicit tau-leaping method with automatic tau selection. *The Journal of Chemical Physics*, 126(22):224101, June 2007. 6.2

[CR08] T. Choi and R. V. Ramamoorthi. Remarks on consistency of posterior distributions. *ArXiv e-prints*, May 2008. 4.3.1, 2, 2, 23, 25

[CSV09] Rohit Chadha, A. Prasad Sistla, and Mahesh Viswanathan. On the expressiveness and complexity of randomization in finite state monitors. *J. ACM*, 56(5), 2009. 2.2

[CY95] Costas Courcoubetis and Mihalis Yannakakis. The complexity of probabilistic verification. *Journal of the ACM*, 42(4):857–907, 1995. 3.1

[dAKN$^+$00] Luca de Alfaro, Marta Z. Kwiatkowska, Gethin Norman, David Parker, and Roberto Segala. Symbolic model checking of probabilistic processes using mtbdds and the kronecker representation. In Susanne Graf and

Michael I. Schwartzbach, editors, *TACAS*, volume 1785 of *Lecture Notes in Computer Science*, pages 395–410. Springer, 2000. 4.5.4

[DCL09] Alexandre Donzé, Gilles Clermont, and Christopher James Langmead. Parameter synthesis in nonlinear dynamical systems: Application to systems biology. In Serafim Batzoglou, editor, *RECOMB*, volume 5541 of *Lecture Notes in Computer Science*, pages 155–169. Springer, 2009. 3.2

[DCL10] Alexandre Donzé, Gilles Clermont, and Christopher James Langmead. Parameter synthesis in nonlinear dynamical systems: Application to systems biology. *J. Comp. Biol.*, 17(3):325–336, 2010. 3.2, 6.2

[DCS$^+$08] Bryan C. Daniels, Yan-Jiun Chen, James P. Sethna, Ryan N. Gutenkunst, and Christopher R. Myers. Sloppiness, robustness, and evolvability in systems biology, May 2008. 3.2

[DFTdJV06] Samuel Drulhe, Giancarlo Ferrari-Trecate, Hidde de Jong, and A. Viari. Reconstruction of switching thresholds in piecewise-affine models of genetic regulatory networks. In João P. Hespanha and Ashish Tiwari, editors, *HSCC*, volume 3927 of *Lecture Notes in Computer Science*, pages 184–199. Springer, 2006. 3.2

[Dij76] Edsger W. Dijkstra. *A Discipline of Programming*. Prentice Hall, Inc., October 1976. 3

[DM07] Alexandre Donzé and Oded Maler. Systematic simulation using sensitivity analysis. In Bemporad et al. [BBB07], pages 174–189. 3.2

[Doo45] J. L. Doob. Markoff chains–denumerable case. *Transactions of the American Mathematical Society*, 58(3):455–473, Nov 1945. 2.1.1

[DtS03] Doron Drusinsky and Man tak Shing. Monitoring temporal logic specifications combined with time series constraints. *J. UCS*, 9(11):1261–1276, 2003. 2.2

[ER05] Kousha Etessami and Sriram K. Rajamani, editors. *Computer Aided Verification, 17th International Conference, CAV 2005, Edinburgh, Scotland, UK, July 6-10, 2005, Proceedings*, volume 3576 of *Lecture Notes in Computer Science*. Springer, 2005. A.2

[Fag05] François Fages. Temporal logic constraints in the biochemical abstract machine biocham. In Patricia M. Hill, editor, *LOPSTR*, volume 3901 of *Lecture Notes in Computer Science*, pages 1–5. Springer, 2005. 2.2

[Fag06] François Fages. From syntax to semantics in systems biology towards automated reasoning tools. 3939:68–70, 2006. 2.2, 3

[FBGH05] James R. Faeder, Michael L. Blinov, Byron Goldstein, and William S. Hlavacek. Rule-based modeling of biochemical networks. *Complexity*, 10(4):22–41, 2005. 1.2, 1.3, 3, 6.2

[FBH05] J. R. Faeder, M. L. Blinov, and W. S. Hlavacek. Graphical rule-based representation of signal-transduction networks. In *SAC '05: Proceedings of the 2005 ACM symposium on Applied computing*, pages 133–140, New York, NY, USA, 2005. ACM. 1.3, 6.2

[FBH08] J. R. Faeder, M. L. Blinov, and W. S. Hlavacek. Rule-based modeling of biochemical systems with BioNetGen. In I. V. Maly, editor, *Systems Biology*, Methods in Molecular Biology. Humana Press, Totowa, NJ, 2008. 1.3, 6.2

[FJK08] Goran Frehse, Sumit Kumar Jha, and Bruce H. Krogh. A counterexample-guided approach to parameter synthesis for linear hybrid automata. In Magnus Egerstedt and Bud Mishra, editors, *HSCC*, volume 4981 of *Lecture Notes in Computer Science*, pages 187–200. Springer, 2008. 3.2

[Flo67] R. W. Floyd. Assigning meanings to programs. In J. T. Schwartz, editor, *Mathematical Aspects of Computer Science, Proceedings of Symposia in Applied Mathematics 19*, pages 19–32, Providence, 1967. American Mathematical Society. 3

[FS01] Bernd Finkbeiner and Henny Sipma. Checking finite traces using alternating automata. In *In Proceedings of Runtime Verification (RV01) [1*, pages 44–60, 2001. 2.2

[GCPD05] R. Gunawan, Y. Cao, L. Petzold, and F. J. Doyle. Sensitivity analysis of discrete stochastic systems. *Biophys J*, 88(4):2530–2540, April 2005. 3.2

[GCSR03] Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. *Bayesian Data Analysis, Second Edition (Texts in Statistical Science)*. Chapman & Hall/CRC, 2 edition, July 2003. 4, 6.1.2

[Gho] Ghosh, editor. *Handbookofsequentialanalysis*. Dekker. 4.4

[Gil75] D. T. Gillespie. The monte carlo method for evaluating integrals. Technical report, National Weapons Center, 1975. 2.1.1

[Gil76] D. T. Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *J. Comp. Phys.*, 22:403–434, 1976. 3

[Gil77] D. T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361, 1977. 5.1, 5.2

[Gil07] Daniel T. Gillespie. Stochastic simulation of chemical kinetics. *Annual review of physical chemistry*, 58(1):35–55, 2007. 5.1, 5.2

[Gir60] I. V. Girsanov. On transforming a certain class of stochastic processes by absolutely continuous substitution of measures. *Theory of Probability and its Applications*, 5(3):285–301, 1960. 4.5.1, 4.5.4, 4.5.5

[Goo99] S. N. Goodman. Toward evidence-based medical statistics. 1: The p value fallacy. *Ann Intern Med*, 130(12):995–1004, June 1999. 3.1.1

[GPS09] Kalpana Gondi, Yogeshkumar Patel, and A. Prasad Sistla. Monitoring the full range of omega-regular properties of stochastic systems. In Neil D. Jones and Markus Müller-Olm, editors, *VMCAI*, volume 5403 of *Lecture Notes in Computer Science*, pages 105–119. Springer, 2009. 2.2

[GS05] Radu Grosu and Scott A. Smolka. Monte carlo model checking. In Nicolas Halbwachs and Lenore D. Zuck, editors, *TACAS*, volume 3440 of *Lecture Notes in Computer Science*, pages 271–286. Springer, 2005. 3.1.1, 4.5

[GTT03] Ronojoy Ghosh, Ashish Tiwari, and Claire Tomlin. Automated symbolic reachability analysis; with application to delta-notch signaling automata. [MP03], pages 233–248. 3.2

[Hay69] G. G. Hays. Computer-aided design: Simulation of digital design logic. *IEEE Trans. Comput.*, 18(1):1–10, 1969. 3

[Hes93] S. L. Heston. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *Review of Financial Studies*, 6:327–343, 1993. 4

[HHMWT00] T. A. Henzinger, B. Horowitz, R. Majumdar, and H. Wong-Toi. Beyond HYTECH: Hybrid systems analysis using interval numerical methods. In *HSCC*, Lecture Notes in Computer Science, pages 130–144. Springer, 2000. 3.2

[HKM08] T. Han, J. P. Katoen, and A. Mereacre. Approximate parameter synthesis for probabilistic time-bounded reachability. In *Proceedings of the IEEE Real-Time Systems Symposium (RTSS 2008), Barcelona, Spain*, pages 173–182, Los Alamitos, December 2008. IEEE Computer Society Press. 3.2, 5.5.2

[HKN$^+$06] J. Heath, M. Kwiatkowska, G. Norman, D. Parker, and O. Tymchyshyn. Probabilistic model checking of complex biological pathways. In C. Priami, editor, *Proc. Computational Methods in Systems Biology (CMSB'06)*, volume 4210 of *Lecture Notes in Bioinformatics*, pages 32–47. Springer Verlag, 2006. 3, 5.6, 5.6.1

[HKN$^+$08] J. Heath, M. Kwiatkowska, G. Norman, D. Parker, and O. Tymchyshyn. Probabilistic model checking of complex biological pathways. *Theoretical Computer Science*, 319(3):239–257, 2008. 3, 5.6, 5.6.1

[HLMP04] T. Hérault, R. Lassaigne, F. Magniette, and S. Peyronnet. Approximate probabilistic model checking. In *Proc. 5th International Conference on Verification, Model Checking and Abstract Interpretation (VMCAI'04)*, volume 2937 of *LNCS*. Springer, 2004. 3.1, 3.1.1

[Hoa69] C. A. R. Hoare. An axiomatic basis for computer programming. *Commun. ACM*, 12(10):576–580, 1969. 3

[Hoe63] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. 3.1.1

[HT08] Espen G. Haug and Nassim N. Taleb. Why we have never used the black-scholes-merton option pricing formula. *Social Science Research Network Working Paper Series*, January 2008. 1.2, 4

[Hul06] J. C. Hull. *Options, Futures, and Other Derivatives*. Prentice-Hall, Upper Saddle River, N.J., sixth edition, 2006. 4

[JBS07] Susmit Jha, Bryan A. Brady, and Sanjit A. Seshia. Symbolic reachability analysis of lazy linear hybrid automata. In Jean-François Raskin and P. S. Thiagarajan, editors, *FORMATS*, volume 4763 of *Lecture Notes in Computer Science*, pages 241–256. Springer, 2007. 5.4.1, 5.5.2

[JCL$^+$09] Sumit Kumar Jha, Edmund M. Clarke, Christopher James Langmead, Axel Legay, André Platzer, and Paolo Zuliani. A bayesian approach to model checking biological systems. In Pierpaolo Degano and Roberto Gorrieri, editors, *CMSB*, volume 5688 of *Lecture Notes in Computer Science*, pages 218–234. Springer, 2009. 1.3, 2.2, 3, 3.1, 4.5, 4.5.2

[Jef61] Harold Jeffreys. *Theory of probability / by Harold Jeffreys*. Clarendon Press, Oxford, 3rd ed. edition, 1961. 1.3, 4.1, 4.1, 4.3

[JJ08] Susmit Jha and Sumit Kumar Jha. Randomization based probabilistic approach to detect trojan circuits. In *HASE*, pages 117–124. IEEE Computer Society, 2008. 4.5

[JKL10] Sumit Kumar Jha, Runpinder Paul Khandpur, and Christopher James Langmead. Model discovery for agent based models using bayesian statistical model checking and abstraction refinement. In *Submitted*, 2010. 6.2

[JKWC07] Sumit Kumar Jha, Bruce H. Krogh, James E. Weimer, and Edmund M.

Clarke. Reachability for linear hybrid automata using iterative relaxation abstraction. In Bemporad et al. [BBB07], pages 287–300. 3.2

[JL10a] Sumit Kumar Jha and Christopher James Langmead. Statistical model checking of stochastic differential equation models using non-i.i.d. sampling. In *Under Review*, 2010. 1.3

[JL10b] Sumit Kumar Jha and Christopher James Langmead. Synthesis and infeasibility analysis for stochastic models of biochemical systems using statistical model checking and abstraction refinement. *Theoretical Computer Science*, page In Press, 2010. 1.3, 3

[JU97] S. Julier and J. Uhlmann. A new extension of the kalman filter to nonlinear systems. In *Int. Symp. Aerospace/Defense Sensing, Simul. and Controls, Orlando, FL*, 1997. 3.2

[Kal60] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960. 3.2

[KD01] Jayesh H. Kotecha and Petar M. Djuric. Gaussian particle filtering. In *Proceedings of the 11th IEEE Signal Processing Workshop on Statistical Signal Processing*, pages 429–432, 2001. 3.2

[KNP04] Marta Z. Kwiatkowska, Gethin Norman, and David Parker. Prism 2.0: A tool for probabilistic model checking. In *QEST*, pages 322–323. IEEE Computer Society, 2004. 2.2, 2.3, 3.1, 4.5.4

[KNP05] M. Kwiatkowska, G. Norman, and D. Parker. Probabilistic model checking in practice: Case studies with PRISM. *ACM SIGMETRICS Performance Evaluation Review*, 32(4):16–21, 2005. 3.1

[Koy90] Ron Koymans. Specifying real-time properties with metric temporal logic. *Real-Time Syst.*, 2(4):255–299, 1990. 2.2

[KR95] Robert E. Kass and Adrian E. Raftery. Bayes factors. *Journal of the American Statistical Association*, 90(430):773–795, 1995. 4.1

[Lan09] C.J. Langmead. Generalized Queries and Bayesian Statistical Model Checking in Dynamic Bayesian Networks: Application to Personalized Medicine. In *Proc. of the 8th International Conference on Computational Systems Bioinformatics (CSB)*, pages 201–212, 2009. 3.1, 3.1.1

[LHFH08] T. Lipniacki, B. Hat, J. R. Faeder, and W. S. Hlavacek. Stochastic effects and bistability in T cell receptor signaling. *J. Theor. Biol.*, page in press, 2008. (document), 1.2, 4.15, 4.3.2, 4.3.2, 4.4.2

[LJ07a] Christopher James Langmead and Sumit Kumar Jha. Predicting protein folding kinetics via temporal logic model checking. In Raffaele Giancarlo and Sridhar Hannenhalli, editors, *Algorithms in Bioinformatics, 7th International Workshop*, volume 4645 of *Lecture Notes in Computer Science*, pages 252–264. Springer, 2007. 2.2, 3.1

[LJ07b] C.J. Langmead and S. K. Jha. Predicting protein folding kinetics via model checking. *Lecture Notes in Bioinformatics: The 7th Workshop on Algorithms in Bioinformatics (WABI)*, pages 252–264, 2007. 3

[LJ09] C.J. Langmead and S.K. Jha. Symbolic Approaches to Finding Control Strategies in Boolean Networks. *J. Bioinf. and Comp. Biol.*, 7(2):323–338, 2009. 2.2

[LJC06] C. J. Langmead, S. Jha, and E. M. Clarke. Temporal-logics as query languages for dynamic bayesian networks: Application to d. melanogaster embryo development, 2006. 2.2

[man63] Manned spacecraft simulation, 1963. 3

[McK95] T.W. McKeithan. Kinetic proofreading in T-cell receptor signal transduction. *Proc Natl Acad Sci*, 92(11):5042–5046, 1995. 4.3.2

[MP03] Oded Maler and Amir Pnueli, editors. *Hybrid Systems: Computation and Control, 6th International Workshop, HSCC 2003 Prague, Czech Republic, April 3-5, 2003, Proceedings*, volume 2623 of *Lecture Notes in Computer Science*. Springer, 2003. A.2

[MS95] H. McAdams and L. Shapiro. Circuit simulation of genetic networks. *Science*, 269:650–656, 1995. 3

[MT00] Ian Mitchell and Claire Tomlin. Level set methods for computation in hybrid systems. In Nancy A. Lynch and Bruce H. Krogh, editors, *HSCC*, volume 1790 of *Lecture Notes in Computer Science*, pages 310–323. Springer, 2000. 3.2

[NRTT09] Aditya V. Nori, Sriram K. Rajamani, SaiDeep Tetali, and Aditya V. Thakur. The Yogi Project: Software property checking via static analysis and testing. In Stefan Kowalewski and Anna Philippou, editors, *TACAS*, volume 5505 of *Lecture Notes in Computer Science*, pages 178–181. Springer, 2009. 1.1

[OL82] Susan S. Owicki and Leslie Lamport. Proving liveness properties of concurrent programs. *ACM Trans. Program. Lang. Syst.*, 4(3):455–495, 1982. 2.2

[Pnu77] Amir Pnueli. The temporal logic of programs. In *FOCS*, pages 46–57. IEEE, 1977. 2.2

[QBdB07] Minh Quach, Nicolas Brunel, and Florence d'Alché Buc. Estimating parameters and hidden variables in non-linear state-space models based on odes for biological networks inference. *Bioinformatics*, 23(23):3209–3216, 2007. 3.2

[RBFS08] Aurélien Rizk, Grégory Batt, François Fages, and Sylvain Soliman. On a continuous degree of satisfaction of temporal logic formulae with applications to systems biology. In *CMSB '08: Proceedings of the 6th International Conference on Computational Methods in Systems Biology*, pages 251–268, Berlin, Heidelberg, 2008. Springer-Verlag. 2.2, 3, 3.2

[RBFS09] Aurélien Rizk, Gregory Batt, François Fages, and Sylvain Soliman. A general computational method for robustness analysis with applications to synthetic gene networks. *Bioinformatics*, 25(12):i169–i178, 2009. 3

[RBL$^+$96] J.D. Rabinowitz, C. Beeson, D. S. Lyonsdagger, M. M. Davisdagger, and H. M. McConnell. Kinetic discrimination in T-cell activation. *Proc Natl Acad Sci*, 93(4):1401–1405, 1996. 4.3.2

[RPCG03] Muruhan Rathinam, Linda R. Petzold, Yang Cao, and Daniel T. Gillespie. Stiffness in stochastic chemically reacting systems: The implicit tau-leaping method. *The Journal of Chemical Physics*, 119(24):12784–12794, 2003. 5.1, 5.2

[SDD$^+$07] David E. Shaw, Martin M. Deneroff, Ron O. Dror, Jeffrey S. Kuskin, Richard H. Larson, John K. Salmon, Cliff Young, Brannon Batson, Kevin J. Bowers, Jack C. Chao, Michael P. Eastwood, Joseph Gagliardo, J. P. Grossman, C. Richard Ho, Douglas J. Ierardi, István Kolossváry, John L. Klepeis, Timothy Layman, Christine McLeavey, Mark A. Moraes, Rolf Mueller, Edward C. Priest, Yibing Shan, Jochen Spengler, Michael Theobald, Brian Towles, and Stanley C. Wang. Anton, a special-purpose machine for molecular dynamics simulation. In *ISCA '07: Proceedings of the 34th annual international symposium on Computer architecture*, pages 1–12, New York, NY, USA, 2007. ACM. 3

[Shr04] Steven E. Shreve. *Stochastic Calculus for Finance II: Continuous-Time Models*, volume 2. Springer Science+Business Media, Inc, 2004. 2.1, 2.1.2, 2.1.2

[SK03] Olaf Stursberg and Bruce H. Krogh. Efficient representation and computation of reachable sets for hybrid systems. In Maler and Pnueli [MP03], pages 482–497. 3.2

[SO96] William Mendenhall Scheaffer, Richard L. and Lyman Ott. *Elementary Survey Sampling*. Duxbury Press, 1996. 1.3, 5.3.1

[SS08] A. Prasad Sistla and Abhigna R. Srinivas. Monitoring temporal properties of stochastic systems. In Francesco Logozzo, Doron Peled, and Lenore D. Zuck, editors, *VMCAI*, volume 4905 of *Lecture Notes in Computer Science*, pages 294–308. Springer, 2008. 2.2

[Sta01] J. Staum. Simulation in financial engineering. In B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, editors, *Proceedings of the 2001 Winter Simulation Conference*, pages 123–133. IEEE Press, 2001. 3

[SVA04] Koushik Sen, Mahesh Viswanathan, and Gul Agha. Statistical model checking of black-box probabilistic systems. In Rajeev Alur and Doron Peled, editors, *CAV*, volume 3114 of *Lecture Notes in Computer Science*, pages 202–215. Springer, 2004. 3.1.1, 4.5

[SVA05] Koushik Sen, Mahesh Viswanathan, and Gul Agha. On statistical model checking of stochastic systems. In *CAV*, LNCS 3576, pages 266–280, 2005. 3.1.1, 4.5

[Tal07] Nassim N. Taleb. *The Black Swan: The Impact of the Highly Improbable*. Random House, April 2007. 6.1.1

[TNO$^+$03] Makoto Taiji, Tetsu Narumi, Yousuke Ohno, Noriyuki Futatsugi, Atsushi Suenaga, Naoki Takada, and Akihiko Konagaya. Protein explorer: A petaflops special-purpose computer system for molecular dynamics simulations. In *SC '03: Proceedings of the 2003 ACM/IEEE conference on Supercomputing*, page 15, Washington, DC, USA, 2003. IEEE Computer Society. 3

[TR05] Prasanna Thati and Grigore Rosu. Monitoring algorithms for metric temporal logic specifications. *Electr. Notes Theor. Comput. Sci.*, 113:145–162, 2005. 2.2

[VAD98] F. Vázquez-Abad and D. Dufresne. Accelerated simulation for pricing Asian options. In *Proceedings of the 1998 Winter Simulation Conference*, pages 1493–1500. IEEE Press, 1998. 3

[vdMDdFW00] Rudolph van der Merwe, Arnaud Doucet, Nando de Freitas, and Eric A. Wan. The unscented particle filter. In Todd K. Leen, Thomas G. Dietterich, and Volker Tresp, editors, *NIPS*, pages 584–590. MIT Press, 2000. 3.2

[Wal47] A. Wald. *Sequential Analysis*. New York: John Wiley and Son, 1947. 3.1.1

[WF00] R. Wu and M. C. Fu. Optimal exercise policies and simulation-based valuation for American-Asian options. manuscript, April 2000. 3

[WW48] A. Wald and J. Wolfowitz. Optimum character of the sequential probability ratio test. *Ann. Math. Statist.*, 19(3):326–339, 1948. 3.1.1

[YBO+98] Bwolen Yang, Randal E. Bryant, David R. O'Hallaron, Armin Biere, Olivier Coudert, Geert Janssen, Rajeev K. Ranjan, and Fabio Somenzi. A performance study of bdd-based model checking. In *FMCAD '98: Proceedings of the Second International Conference on Formal Methods in Computer-Aided Design*, pages 255–289, London, UK, 1998. Springer-Verlag. 3.1

[YKNP04] Håkan L. S. Younes, Marta Z. Kwiatkowska, Gethin Norman, and David Parker. Numerical vs. statistical probabilistic model checking: An empirical study. In Kurt Jensen and Andreas Podelski, editors, *TACAS*, volume 2988 of *Lecture Notes in Computer Science*, pages 46–60. Springer, 2004. 4.5

[YKNP06] Håkan L. S. Younes, Marta Z. Kwiatkowska, Gethin Norman, and David Parker. Numerical vs. statistical probabilistic model checking. *STTT*, 8(3):216–228, 2006. 2.3, 3.1.1, 4.5

[You04] Hakan Lorens Samir Younes. *Verification and planning for stochastic processes with asynchronous events*. PhD thesis, Pittsburgh, PA, USA, 2004. Chair-Reid G. Simmons. 2.2, 2.3, 3.1, 3.1.1, 3.1.1, 4.5, 6.1.1

[You05a] Håkan L. S. Younes. Probabilistic verification for "black-box" systems. In Etessami and Rajamani [ER05], pages 253–265. 4.5

[You05b] Håkan L. S. Younes. *Verification and Planning for Stochastic Processes with Asynchronous Events*. PhD thesis, Carnegie Mellon, 2005. 4.5

[You05c] Håkan L. S. Younes. Ymer: A statistical model checker. In Etessami and Rajamani [ER05], pages 429–433. 4.5

[YS02] Håkan L. S. Younes and Reid G. Simmons. Probabilistic verification of discrete event systems using acceptance sampling. In *CAV*, LNCS 2404, pages 223–235. Springer, 2002. 2.3, 3.1, 3.1.1, 4.5

[YS06] Håkan L. S. Younes and Reid G. Simmons. Statistical probabilistic model checking with a focus on time-bounded properties. *Information and Computation*, 204(9):1368–1409, 2006. 2.2, 2.3, 3.1