# Statistical Model Checking for Complex Stochastic Models in Systems Biology

**Sumit Kumar Jha** **Edmund M. Clarke**

**Christopher James Langmead** **Axel Legay**

**Andre Platzer** **Paolo Zuliani**

March 19, 2009

CMU-CS-09-110
School of Computer Science

Carnegie Mellon University

Pittsburgh, PA 15213

This research was sponsored by the GSRC (University of California) under contract no. SA423679952, National Science Foundation under contracts no. CCF0429120, no. CNS0411152, and no. CCF0541245, Semiconductor Research Corporation under contract no. 2005TJ1366, Air Force (University of Vanderbilt) under contract no. 18727S3, International Collaboration for Advanced Security Technology of the National Science Council, Taiwan, under contract no. 1010717, the U.S. Department of Energy Career Award (DE-FG02-05ER25696), and a Pittsburgh Life-Sciences Greenhouse Young Pioneer Award.

**Keywords:** Model Checking, Temporal Logic, Stochastic Systems, Statistical Hypothesis Testing, Systems Biology, Biochemical Pathways, Parameter Synthesis

**Abstract**

Recently, there has been considerable interest in the use of Model Checking for Systems Biology. Unfortunately, the state space of stochastic biological models is often too large for classical Model Checking techniques. For these models, a statistical approach to Model Checking has been shown to be an effective alternative. Extending our earlier work, we present the first algorithm for performing statistical Model Checking using Bayesian Sequential Hypothesis Testing. We show that our Bayesian approach outperforms current statistical Model Checking techniques, which rely on tests from Classical (aka Frequentist) statistics, by requiring fewer system simulations. Another advantage of our approach is the ability to incorporate prior Biological knowledge about the model being verified. We demonstrate our algorithm on a variety of models from the Systems Biology literature and show that it enables faster verification than state-of-the-art techniques, even when no prior knowledge is available.

# 1 Introduction

Computational models are increasingly used in the field of Systems Biology to examine the dynamics of biological processes (e.g., [1, 9, 11, 21, 30, 33, 36]). By 'computational', we mean discrete-variable and continuous or discrete-time models [5], where the components of the system interact and evolve by obeying a set of instructions or rules. In contrast to differential equation-based models, which are also widely used in Systems Biology, computational models can provide insights into the role of stochastic effects over discrete-populations of molecules or cells. Recently, there has been considerable interest in the application of Model Checking [14] as a powerful tool for *formally* reasoning about the dynamic properties of such models (e.g., [2, 7, 10, 12, 16, 19, 25, 37]). This paper presents a new Model Checking algorithm that is well-suited for verifying properties of very large stochastic models, such as those created and used in Systems Biology.

The stochastic nature of most computational models from Systems Biology gives rise to an instance of the *Probabilistic Model Checking* (PMC) problem [15, 14, 31]. Suppose $\mathcal{M}$ is a stochastic model over a set of states $S$, $s_0$ is a starting state, $\phi$ is a dynamic property expressed as a formula in temporal logic, and $\theta \in [0,1]$ is a probability threshold. The PMC problem is: given the 4-tuple $(\mathcal{M}, s_0, \phi, \theta)$, algorithmically decide whether $\mathcal{M}, s_0 \models P_{\geqslant\theta}(\phi)$. In this paper, property $\phi$ is expressed in BLTL - Bounded Linear Temporal Logic [35, 34, 20]. Given these, PMC algorithms decide whether the model satisfies the property with at least probability $\theta$.

Existing algorithms for solving the PMC problem fall into one of two categories. The first category comprises numerical methods (e.g. [3, 4, 13, 17, 31]) which can compute the probability with which the property holds with high precision. Numerical methods are generally only suitable for small systems ($\approx 10^6$ to $10^7$ states). In a Biological System, the number of states can easily exceed this limit, which motivates the need for algorithms for solving the PMC problem in an approximate fashion. Approximate methods (e.g., [24, 27, 38, 45]) work by sampling a set of *traces* from the model. Each trace is then evaluated to determine whether it satisfies the property.

The number of satisfying traces is used to (approximately) decide whether $\mathcal{M}, s_0 \models P_{\geqslant\theta}(\phi)$.

Approximate PMC methods can be further divided into two sub-categories: (i) those that seek to *estimate* the probability that the property holds and then compare that estimate to $\theta$ (e.g., [27, 38]), and (ii) those that reduce the PMC problem to a *hypothesis testing* problem (e.g., [45, 46]). That is, deciding between two hypotheses — $H_0 : P_{\geqslant\theta}(\phi)$ versus $H_1 : P_{<\theta}(\phi)$. Hypothesis-testing based methods are more efficient than those based on estimation when $\theta$ (which is specified by the user) is significantly different than the true probability that the property holds (which is determined by $\mathcal{M}$ and $s_0$) [44].

Existing PMC methods based on hypothesis testing rely on *Classical* (aka *Frequentist*) statistical procedures, like Wald's Sequential Probability Ratio Test (SPRT) [41], to answer the decision problem. Our algorithm performs hypothesis testing, but uses *Bayesian* statistical procedures. This distinction is not trivial, as Bayesian and Classical statistics are two very different fields. We will show that in practice, our Bayesian approach requires fewer samples than Wald's SPRT. Finally, we note that because we adopt a Bayesian approach, our algorithm can incorporate prior knowledge, in the form of a probability distribution, $P(\theta)$, when available. This is relevant because in a Biological setting, it is often the case that prior knowledge is available.

The contributions of this paper are as follows:

- The first application of Bayesian Sequential Hypothesis Testing to statistical Model Checking,
- The first hypothesis-testing based statistical Model Checking algorithm designed for composite hypotheses, which can in particular include prior knowledge via a mixture of prior distributions,
- A theorem proving that our algorithm terminates with probability 1,
- Error bounds for our algorithm, and
- A series of case studies using Systems Biology models demonstrating that our method is empirically more efficient than existing algorithms for statistical Model Checking.

# 2 Background and Related Work

Our algorithm can be applied to any stochastic model $\mathcal{M}$ with a well-defined probability space over traces. Several well-studied stochastic models like (discrete and continuous) Markov Chains satisfy this property [46]. We assume that each execution of the system can be represented by a sequence of states and the time spent in these states. The sequence $\sigma = (s_0,t_0),(s_1,t_1),\ldots$ denotes an execution of the system along states $s_0,s_1,\ldots$ with durations $t_0,t_1,\ldots \in \mathbb{R}$. The system stays in state $s_i$ for duration $t_i$ and makes a transition to $s_{i+1}$. We require that the sum $\sum_i^\infty t_i$ must diverge, that is, the system can not make infinitely many state switches in finite time.

## 2.1 Specifying Properties in Temporal Logic

Our algorithm verifies properties of $\mathcal{M}$ expressed as formulas in *Probabilistic Bounded Linear Temporal Logic* (PBLTL). We first define the syntax and semantics of *Bounded Linear Temporal Logic* (BLTL) [35, 34, 20] and then extend that logic to PBLTL.

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

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

where $\sim \in \{\geqslant,\leqslant,=\}$, $x \in SV$, $v \in \mathbb{Q}$, 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 $\mathbf{U^t}$.

We define the semantics of BLTL with respect to executions of $\mathcal{M}$. The fact that an execution $\sigma$ satisfies property $\phi$ is denoted by $\sigma \models \phi$. Let $\sigma = (s_0,t_0),(s_1,t_1),\ldots$ be an execution of the model along states $s_0,s_1,\ldots$ with durations $t_0,t_1,\ldots \in \mathbb{R}$. We denote the execution trace starting at state

$i$ by $\sigma^i$ (in particular, $\sigma^0$ denotes the original execution $\sigma$). The value of the state variable $x$ in $\sigma$ at the state $i$ is denoted by $V(\sigma,i,x)$. The semantics of BLTL for a trace $\sigma^k$ starting at the $k^{th}$ state ($k \in \mathbb{N}$) 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 ($a$) $\sum_{0 \leqslant l < i} t_{k+l} \leqslant t$, ($b$) $\sigma^{k+i} \models \phi_2$ and ($c$) for each $0 \leqslant j < i$, $\sigma^{k+j} \models \phi_1$.

Statistical Model Checking is based on evaluating whether $\sigma \models \phi$ holds on sample simulations $\sigma$ of the system. In practice, sample simulations only have a finite duration. The question is how long these simulations have to be for the formula $\phi$ to have a well-defined semantics such that $\sigma \models \phi$ can be checked. If $\sigma$ is too short, say of duration 2, the semantics of $\phi_1 \mathbf{U}^5 \phi_2$ may be unclear. But at what duration of the simulation can we stop because we know that the truth-value for $\sigma \models \phi$ will never change by continuing the simulation? In Appendix A, we prove that finite simulations of bounded duration are always sufficient for Model Checking BLTL on traces.

We can now define Probabilistic Bounded Linear Temporal Logic.

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

We say that $\mathcal{M}$ satisfies PBLTL property $P_{\geqslant\theta}(\phi)$, denoted by $\mathcal{M} \models P_{\geqslant\theta}(\phi)$, if and only if the probability that an execution of $\mathcal{M}$ satisfies BLTL property $\phi$ is greater than or equal to $\theta$. The problem is well-defined [46] since one can always assign a unique probability measure to the set of executions of $\mathcal{M}$ that satisfy a formula in BLTL. Note that counterexamples to the BLTL property $\phi$ are *not* counterexamples to the PBLTL property $P_{\geqslant\theta}(\phi)$, because the truth of $P_{\geqslant\theta}(\phi)$ depends on

the likelihood of all counterexamples to $\phi$. This makes PMC more difficult than standard Model Checking, because one counterexample to $\phi$ is not enough to answer $P_{\geqslant\theta}(\phi)$.

## 2.2 Existing Statistical Probabilistic Model Checking Algorithms

As outlined in the introduction, Probabilistic Model Checking algorithms can either be exact (e.g. [3, 4, 13, 17, 31]), or statistical in nature. In practice, statistical methods (e.g., [24, 27, 38, 45]), which iteratively draw sample traces from the model, are generally better suited to Model Checking Biological systems because they scale better. Our method is statistical, and so we will compare and contrast our method to existing statistical methods in this section.

Existing PMC methods based on hypothesis testing rely on *Classical* (aka *Frequentist*) statistical procedures, like Wald's Sequential Probability Ratio Test (SPRT) [41], to answer the decision problem. Younes and Simmons introduced the first algorithm for statistical Model Checking [44, 45, 46] for verifying probabilistic temporal properties of stochastic systems. Their work uses the SPRT, which is designed for *simple* hypothesis testing[1]. Specifically, the SPRT decides between the simple null hypothesis $H_0' : \mathcal{M}, s_0 \models P_{=\theta_0}(\phi)$ against the simple alternate hypothesis $H_1' : \mathcal{M}, s_0 \models P_{=\theta_1}(\phi)$, where $\theta_0 < \theta_1$. 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 the tests satisfying the same Type I and II errors [42], when either $H_0'$ or $H_1'$ is true. The PMC problem is instead a choice between two *composite* hypotheses $H_0 : \mathcal{M}, s_0 \models P_{\geqslant\theta}[\phi]$ versus $H_1 : \mathcal{M}, s_0 \models P_{<\theta}[\phi]$. The SPRT is not defined unless $\theta_0 \neq \theta_1$, so Younes and Simmons overcome this problem by separating the two hypotheses by an *indifference region* $(\theta-\delta, \theta+\delta)$, where $0 < \delta < 1$ is a user-specified parameter. It can be shown that the SPRT with indifference region can be used for testing composite hypotheses, while respecting the same Type I and II errors of a standard SPRT [22, Section 3.4].

[1] A simple hypothesis completely specifies a distribution. For example, a Bernoulli distribution of parameter $p$ is fully specified by the hypothesis $p = 0.5$ (or some other fixed value). A composite hypothesis has instead free parameters, *e.g.* the hypothesis $p < 0.3$, for a Bernoulli distribution.

However, in this case the test is no longer optimal, and the maximum expected sample size may be much bigger than the optimal fixed sample size sampling test - see [8] and [22, Section 3.6]. We note that our algorithm solves the composite hypothesis testing problem, but does so using Bayesian statistics, and thus requires no indifference region.

The method of [27] uses a fixed number of samples and estimates the probability the property holds as the number of satisfying traces divided by the number of sampled traces. Their algorithm guarantees the accuracy of the results using Chernoff-Hoeffding bounds. In particular, their algorithm can guarantee that the difference in the estimated and the true probability is less than $\varepsilon$, with probability $\rho$, where $\rho < 1$ and $\varepsilon > 0$ are user-specified parameters. Grosu and Smolka use a similar technique for verifying formulas in LTL [24]. Their algorithm randomly samples lassos from a Büchi automaton in an on-the-fly fashion.

Finally, Sen *et al.* [38, 39] used the *p-value* for the null hypothesis as a 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. Sen *et al.*'s method does not have a way to control the Type I and II errors.

# 3 Bayesian Statistical Model Checking

In this section, we first review some important concepts from statistical Model Checking, and then introduce theory and terminology from Bayesian statistics. We then present our algorithm in Sec. 3.2.

Recall that the PMC problem is to decide whether $\mathcal{M} \models P_{\geqslant\theta}(\phi)$, where $\theta \in (0,1)$ and $\phi$ is a BLTL formula. Let $p$ be the (unknown but fixed) probability of the model satisfying $\phi$: thus, the PMC problem can now be stated as deciding between two hypotheses:

$$H_0 : p \geqslant \theta \qquad H_1 : p < \theta.$$

For any trace $\sigma_i$ of the system, we can deterministically decide whether $\sigma_i$ satisfies $\phi$. Therefore, we can define a Bernoulli random variable $X_i$ denoting the outcome of $\sigma_i \models \phi$. The probability mass function associated with $X_i$ is thus:

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

where $x_i = 1$ iff $\sigma_i \models \phi$, otherwise $x_i = 0$. Note that the $X_i$ are independent and identically distributed. Since $p$ is unknown, we assume that it is given by a random variable, whose density $g(\cdot)$ is called the *prior* density. The prior 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 formula is usually summarized by a *non-informative* or *objective* prior probability.

## 3.1 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. Then Bayes' theorem states that the *posterior odds* are

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

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 odds for hypotheses $H_0$ and $H_1$ given data $d$ is

$$\frac{P(H_0|d)}{P(H_1|d)} = \frac{P(d|H_0)}{P(d|H_1)} \frac{P(H_0)}{P(H_1)} \ . \tag{1}$$

**Definition 2** *The Bayes factor $\mathcal{B}$ of sample $d$ and hypotheses $H_0$ and $H_1$ is*

$$\mathcal{B} = \frac{P(d|H_0)}{P(d|H_1)} .$$

For fixed priors in a given example, the Bayes factor is directly proportional to the posterior odds ratio by Equation (1). Thus, it may be used as a measure of relative confidence in $H_0$ vs. $H_1$, as proposed by Jeffreys [29]. In particular, he suggested that a value of the Bayes factor greater than 100 provides decisive evidence in favor of $H_0$. To test $H_0$ vs. $H_1$ we compute the Bayes factor $\mathcal{B}$ of the available data and then compare it against a fixed threshold $T > 1$: we shall accept $H_0$ iff $\mathcal{B} > T$. Jeffreys interpretes the value of the Bayes factor as a measure of the evidence in favor of $H_0$ (dually, $\frac{1}{\mathcal{B}}$ is the evidence in favor of $H_1$).

We now show how to compute the Bayes factor. According to Definition 2, we have to calculate the probability of the observed sample $d = (x_1, \ldots, x_n)$ given $H_0$ and $H_1$. They are given by integrating the joint density $h(d|\cdot)$ with respect to the prior $g(\cdot)$, and since we assume that the sample is drawn from iid variables, we have that $h(d|\cdot) = f(x_1|\cdot) \cdots f(x_n|\cdot)$. Therefore, the Bayes factor is the ratio:

$$\mathcal{B} = \frac{P(x_1, \ldots, x_n|H_0)}{P(x_1, \ldots, x_n|H_1)} = \frac{\int_\theta^1 f(x_1|u) \cdots f(x_n|u) \cdot g(u)\, du}{\int_0^\theta f(x_1|u) \cdots f(x_n|u) \cdot g(u)\, du} . \tag{2}$$

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 $g$. Hence, the choice of the threshold Bayes Factor ($T$) in Sec. 3.2 also indicates an objective degree of confidence in the accepted hypothesis when the Bayesian Statistical Model Checking algorithm stops.

## 3.2 Algorithm:

Our algorithm is essentially a sequential version of Jeffreys' test. Remember we want to establish whether $\mathcal{M} \models P_{\geqslant\theta}(\phi)$, where $\theta \in (0,1)$ and $\phi$ is a BLTL formula. Like all statistical Model Checking algorithms, we assume that it is possible to generate unbiased samples from the model. The algorithm iteratively draws independent and identically distributed sample traces $\sigma_1, \sigma_2, ...$, and checks whether they satisfy $\phi$. As explained above, we can model this procedure as independent sampling from a Bernoulli distribution $X$ of unknown parameter $p$ - the actual probability of the model satisfying $\phi$. At stage $n$ the algorithm has drawn samples $x_1, \ldots, x_n$ iid like $X$. It then computes the Bayes factor $\mathcal{B}_n$ according to (2), and it stops iff $(\mathcal{B}_n > T \ \vee \ \mathcal{B}_n < \frac{1}{T})$. When this occurs, it will accept $H_0$ iff $\mathcal{B}_n > T$, and will accept $H_1$ iff $\mathcal{B}_n < \frac{1}{T}$. The algorithm is shown below.

**Algorithm 1** Bayesian Statistical Model Checking

**Require:** PBLTL Property $P_{\geqslant\theta}(\phi)$, Threshold $T > 1$, Prior density $g$ for unknown parameter $p$

```
n := 0        {number of traces drawn so far}
x := 0        {number of traces satisfying φ so far}
repeat
   σ := draw a sample trace of the system (iid)
   n := n + 1
   if σ ⊨ φ then
      x := x + 1
   end if
   B_n := BayesFactor(n, x)        {compute according to Equation (2)}
until (B_n > T ∨ B_n < 1/T)
if (B_n > T) then
   return H_0 accepted
else
   return H_1 accepted
end if
```

From (2) we see that the algorithm can incorporate prior knowledge through $g$, when computing the Bayes factor. Our examples focus on Beta priors which are defined over the $(0,1)$ interval by

the following probability density (for real parameters $\alpha, \beta > 0$):

$$\forall u \in (0,1) \quad g(u,\alpha,\beta) \mathrel{\widehat{=}} \frac{1}{B(\alpha,\beta)} u^{\alpha-1}(1-u)^{\beta-1} \tag{3}$$

where the Beta function $B(\alpha,\beta)$ is defined as:

$$B(\alpha,\beta) \mathrel{\widehat{=}} \int_0^1 t^{\alpha-1}(1-t)^{\beta-1} dt \ . \tag{4}$$

By varying the parameters $\alpha$ and $\beta$, one can approximate other smooth unimodal densities on $(0,1)$ by a Beta density (*e.g.* the uniform density over $(0,1)$ is a Beta with $\alpha = \beta = 1$). We also define the Beta distribution function $F_{(\alpha,\beta)}(u)$:

$$\forall u \in (0,1) \quad F_{(\alpha,\beta)}(u) \mathrel{\widehat{=}} \int_0^u g(t,\alpha,\beta)\, dt = \frac{1}{B(\alpha,\beta)} \int_0^u t^{\alpha-1}(1-t)^{\beta-1}\, dt \tag{5}$$

which is just the usual distribution function for a Beta random variable of parameters $\alpha, \beta$ (*i.e.*, the probability that it takes values less than or equal to $u$).

The choice of the Beta density is not arbitrary. It is well-known that the Beta distribution is the *conjugate prior* to the Bernoulli distribution[2]. This relationship gives rise to closed-form solutions to the *posterior* density over $\theta$ (i.e., $P(\theta|d)$), thus avoiding numerical integration when calculating the Bayes factor. Our data $(x_1,\ldots,x_n)$ are assumed to be iid samples drawn from a Bernoulli distribution of unknown parameter $p$. We write $x = \sum_{i=1}^n x_i$ for the number of successes in $(x_1,\ldots,x_n)$. The prior density $g(\cdot)$ is assumed to be a Beta density with fixed parameters $\alpha, \beta > 0$. In Appendix B we show that the Bayes factor $\mathcal{B}_n$ at stage $n$ can be computed in terms of the Beta distribution function:

$$\mathcal{B}_n = \frac{1}{F_{(x+\alpha,n-x+\beta)}(\theta)} - 1 \ .$$

[2]A distribution $P(\theta)$ is said to be a conjugate prior for a likelihood function, $P(d|\theta)$, if the posterior, $P(\theta|d)$ is in the same family of distributions.

The Beta distribution function can be computed with high accuracy by standard mathematical libraries (*e.g.* the GNU Scientific Library) or software (*e.g.* Matlab). Hence, the Beta distribution is the appropriate choice for summarizing the prior probability distribution in Statistical Model Checking. We present the following two Theorems:

**Theorem 1 (Termination)** *The Bayesian Statistical Model Checking algorithm terminates with probability one, for Beta priors and Bernoulli samples. (See Appendix C for a proof.)*

**Theorem 2** *If the Bayesian Model Checking algorithm terminates after observing n sample traces, an upper bound on the probability of the Type I error is*

$$\sum_{x=0}^{n} I_{\{\mathcal{B}(n,x) < 1/T\}}(x) \binom{n}{x} t_{max}^{x}(1-t_{max})^{n-x}$$

*where $t_{max}$ is the value of $t$ that maximizes the expression $t^i(1-t)^{n-i}$ defined on $[\theta,1]$, $T$ is the Bayes Factor threshold used in the Bayesian Model Checking algorithm, and $I$ is the indicator function. (See Appendix D for a proof.)*

## 3.3 Verification Over General Priors

The use of conjugate priors does not pose restrictions, in practice. It is known that any prior distribution (with or without a density) can be well approximated by a *finite* mixture of conjugate priors [18]. Thus, we can approximate an arbitrary prior over $(0,1)$ by constructing a density $G(\cdot)$ of the form:

$$G(u) \widehat{=} \sum_{i=1}^{N} r_i \cdot g_i(u, \alpha_i, \beta_i)$$

where $N$ is a positive integer which depends on the level of accuracy required, the $g_i$'s are Beta densities (of possibly different parameters $\alpha_i, \beta_i$), and the $r_i$'s are positive reals summing up to 1 - this ensures that $G$ is a proper density.

For such priors, the computation of the Bayes factor is slightly more complicated. In Appendix B we show that the Bayes factor at stage $n$ is given by:

$$\mathcal{B}_n = \frac{\sum_{i=1}^{N} r_i' \cdot B(x+\alpha_i, n-x+\beta_i)}{\sum_{i=1}^{N} r_i' \cdot B(x+\alpha_i, n-x+\beta_i) \cdot F_{(x+\alpha_i, n-x+\beta_i)}(\theta)} - 1$$

where $r_i' = \frac{r_i}{B(\alpha_i,\beta_i)}$. Again, we see that the Bayes factor can be computed by means of standard, well-known numerical methods, thereby simplifying the implementation of the algorithm. Theorem 1 can be extended to handle this case, too (See Appendix C.2).

# 4 Benchmarks

In this section, we analyze the performance of our algorithm on five benchmark models from the Systems Biology literature. Three of the models are written in the PRISM Model Checking tool's specification language [28, 31], and the remaining two are written in SBML and were obtained from the Matlab Systems Biology Toolbox. The PRISM Model Checker tool is capable of both symbolic (i.e., exact) Probabilistic Model Checking, and statistical Probabilistic Model Checking. PRISM's statistical Probabilistic Model Checking Algorithm implements the algorithm of [27] which uses a fixed sized sampling approach and estimates the true probability as the number of satisfying traces over the number of sampled traces. We note that for each of the benchmark sets, we consider models that are too large for symbolic model checking.

Our experiments demonstrate two important properties of our algorithm: (i) we show that our algorithm requires fewer traces than either the algorithm of [27] implemented in PRISM or Wald's SPRT algorithm - while retaining the same bounds on the frequentist Type-I and Type-II error probabilities. (ii) The performance of both the Wald's algorithm [41] and our Bayesian Model Checking algorithm degrades as the threshold probability (i.e., $\theta$) in the PBLTL temporal logic formula gets close to the actual probability of the model satisfying the BLTL formula. However,

the Bayesian algorithm shows a more graceful degradation compared to the Wald's SPRT approach.

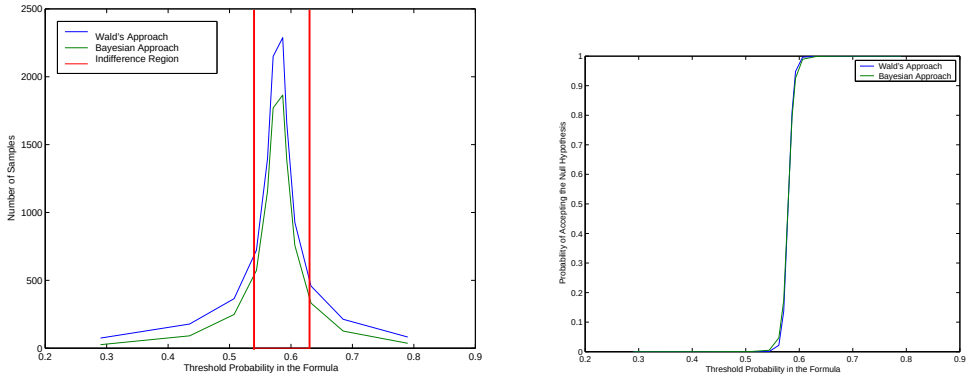
## 4.1 PRISM Benchmarks

We studied three large PRISM benchmarks which are not well suited for numerical approaches to Probabilistic Model Checking. In our experiments, the Bayesian Model Checking algorithm used uniform priors, and accepted a hypothesis when it was 10000 times more likely than the other hypothesis (Bayes Factor threshold $T = 10000$). Our experiments with Wald's SPRT used Type I and II error bounds of 0.0001. We chose an indifference region $\delta$ so as to make the Type I and Type II errors for both the Wald's Test and the Bayes Factor test equal. The statistical estimation engine of the PRISM model checker always needed 92042 samples to estimate the probability of the BLTL formulae being true.

The results of experiments with the Fibroblast Growth Factor Signalling Model (see [25], [26] for details) are presented. We checked the property whether the probability that Grb2 binds to FRS2 within 20 time units exceeds $\theta$ (for several values of $\theta$):

$$H_0 : \mathcal{M} \models P_{\geqslant \theta}[\ \mathbf{F^{20}}\ (FRS2\_GRB > 0\ )]$$

The power curves and the number of samples for this benchmark are plotted in Fig. 2(a) and Fig. 2(b) respectively. A power curve indicates the probability of accepting the null hypothesis for various values of the threshold probability $\theta$ in the PBLTL formula. We chose the Wald's Test so that its power curve matched that of the Bayesian Test at the 0.0001 and 0.9999 acceptance probability. The goal is to make sure that the two tests have equal statistical power. From Figure 2(b), it is clear that both the power curves are almost on top of each other and hence, both the tests have indeed been calibrated to be equally powerful. The Bayesian algorithm needs fewer samples than the Wald's SPRT test for this benchmark. This shows that the Bayesian Statistical Model Checking




(a) Number of Samples for various probability thresholds in the formula.

(b) Power Curve of the Bayesian and Wald's approach.

Figure 1: **Fibroblast Growth Factor Signalling Model:** The system satisfies the formula with probability 0.58. (Bayes Factor=10000)

performs better than an approach based on Wald's SPRT.

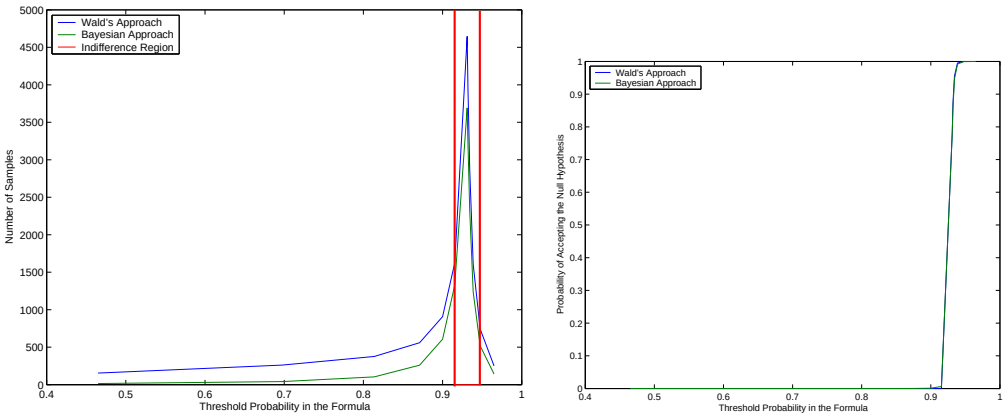
We also studied the continuous time Markov Chain model [6, 40] for circadian rhythm. We checked the property that the probability of the number of activated messenger RNAs exceeding 5 units within 0.25 time units is more than θ (for various values of θ):

$$H_0 : \mathcal{M} \models P_{\geqslant \theta}[\, \mathbf{F}^{\mathbf{0.25}}\ (ma > 5)\ ]$$

The power curves and the number of samples for this benchmark are plotted in Fig. 2(b) and Fig. 2(a) respectively. We calibrated the Wald's Test so that its power curve closely matched that of the Bayesian Test so as to make a fair comparison. From the figure, we observe that the Bayesian algorithm always needs fewer samples than the Wald's SPRT test for this benchmark.

We also analyzed the model on Cell cycle control [32] and studied the probability that Cyclin gets bound within the first 0.5 time units. We check the property that the probability of the number of bound Cyclin molecules exceeds 3 units within 0.5 time units exceeds θ (for various values of θ):

$$H_0 : \mathcal{M} \models P_{\geqslant \theta}[\, \mathbf{F}^{\mathbf{0.5}}\ (cyclin\_bound > 3)\ ]$$




(a) Number of Samples for various probability thresholds in the formula.

(b) Power Curve of the Bayesian and Wald's approach.

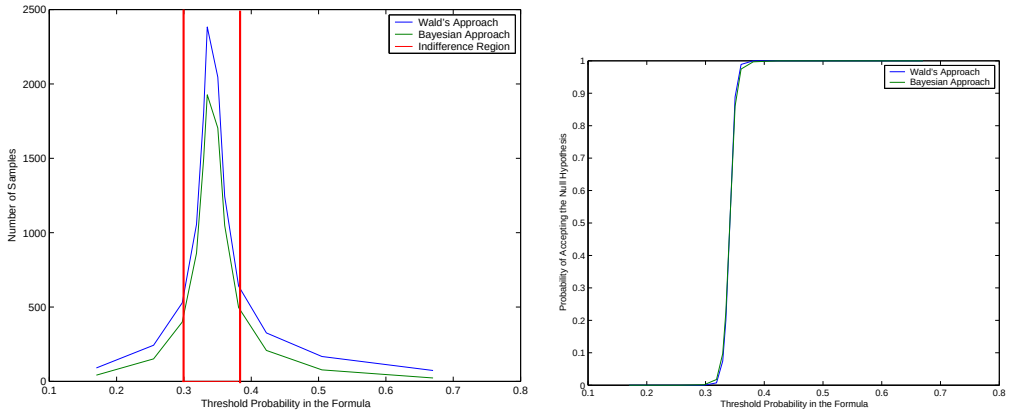
Figure 2: **Circadian Rythm:** The system satisfies the formula with probability 0.93. (Bayes Factor=10000)

The results of our experiment are presented in Fig. 3(a). The Bayesian Statistical Model Checking algorithm usually required fewer samples than the approach based on Wald's SPRT.

## 4.2 SBML Experiments

We also studied SBML models using the implementation of Gillespie's Stochastic Simulation Algorithm in Matlab's Systems Biology Toolbox. We analyzed two large models with over $10^8$ and $10^{17}$ species. We used monitors written in Matlab to verify the BLTL properties on traces. Our analysis of the experiments in this section is purely Bayesian, i.e., we have studied the performance of the algorithm over only one run (using uniform priors). In the previous sections, we had compared the performance of our algorithm with Wald's SPRT by running the algorithm several times on the same model - a frequentist approach.

We analyzed the Yeast Heterotrimeric G Protein Cycle benchmark [43]. We analyzed the property that the G protein stays above the threshold of 6000 units for 2 time units and falls below




(a) Number of Samples for various probability thresholds in the formula.

(b) Power Curve of the Bayesian and Wald's approach.

Figure 3: **Cell Cycle Control:** The system satisfies the formula with probability 0.34. (Bayes Factor=10000)

6000 before 20 time units.

$$H_0 : \mathcal{M} \models P_{\geqslant \theta}[\, \mathbf{G}^2(GProtein < 6000) \text{ and } \mathbf{F}^{20}(GProtein > 6000)] \,.$$

We also ran experiments using the Lotka model [23] and verified the property that the number of copies of the $x$ species rises to a threshold level within 0.01 time units.

$$H_0 : \mathcal{M} \models P_{\geqslant \theta}[\, \mathbf{F}^{0.01}(x > 1.4 * 10^7)]$$

The results of our experiments are shown in Table 1: both hypotheses are always accepted, although the number of samples increases with the probability threshold of the temporal formula.

| Probability | # Samples Needed |
|---|---|
| 0.2 | 3 |
| 0.6 | 8 |
| 0.8 | 14 |
| 0.9 | 23 |
| 0.9999 | 99 |

| Probability | # Samples Needed |
|---|---|
| 0.1 | 2 |
| 0.5 | 6 |
| 0.7 | 10 |
| 0.9 | 23 |
| 0.99 | 69 |

Table 1: Performance on the G Protein (left) and Lotka Benchmark (right). (Bayes Factor = 100)

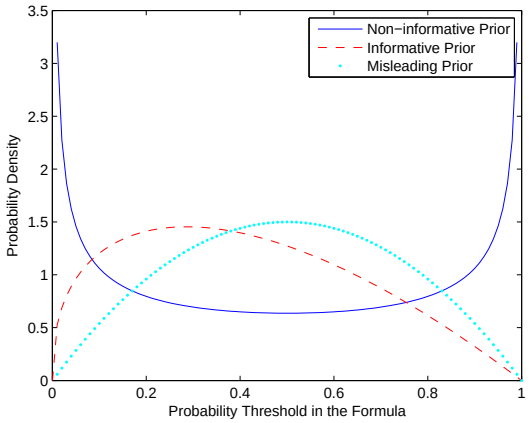
### 4.3 Experiment with Different Classes of Priors

We investigated the effect of priors on the performance of the Bayesian Model Checking algorithm. We used three different priors - non-informative prior, an informative prior and a misleading prior. The priors, the number of samples needed by the Bayesian algorithm for these priors, and the power curve for each of these priors is also plotted in Fig. 4(a), Fig. 4(b) and Fig. 4(c) respectively. The priors used are Beta distributions with different shape parameters: (i) $\alpha = 1/2, \beta = 1/2$: non-informative prior, (ii) $\alpha = 1.4, \beta = 2$ : informative prior with a peak around 0.34 (iii) $\alpha = 2, \beta = 2$: a misleading prior with peak around 0.5.

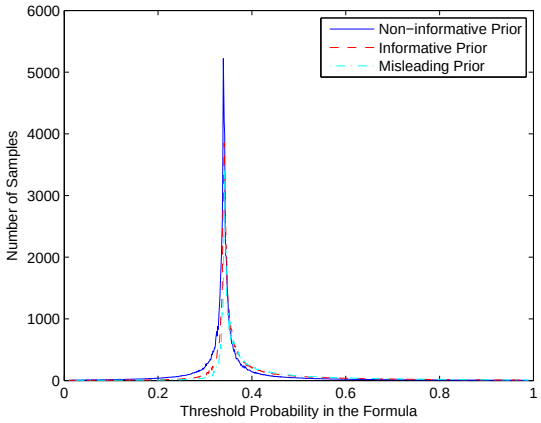
Fig. 4(b) shows that the number of samples needed by the Bayesian algorithm becomes smaller when the prior probability distribution is informative and supports the true hypothesis. Also, the power curve (see Fig. 4(c)) becomes sharper when the Bayesian algorithm is given a correct and informative prior probability distribution. A completely non-informative prior also performs well both in the number of samples and the power of the test. Strongly misleading priors make the power curve less steep. However, the algorithm still performs quite well when the actual probability of the system is away from the threshold probability in the formula.

## 5 Conclusions and Future Work

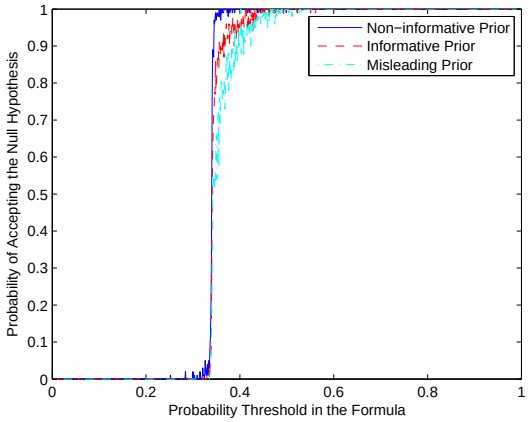
We have introduced the first algorithm for Probabilistic Model Checking based on Bayesian Sequential Hypothesis Testing. Our algorithm terminates with probability 1, and provides bounds on the probability of returning an incorrect answer. Empirically, we have shown that our algorithm requires fewer traces to terminate than techniques based on Classical Statistics. This is not surprising as the Bayesian method comparing composite hypotheses whereas techniques like Wald's SPRT are comparing simple hypotheses. This advantage in efficiency is important in the context of Systems Biology as the cost of generating traces is not necessarily negligible. Bayesian methods also




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




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




(c) Power curve of the tests of the Algorithm.

Figure 4: **Different Classes of Priors**

afford a convenient means for incorporating domain knowledge through the prior distributions.

Our algorithm is presently limited to incorporating prior information on the probability that the property is true. A more fully Bayesian approach would incorporate prior information on not just the property, but also the starting state and parameters of the model. We are presently extending our method to address this limitation.

## Acknowledgements

The authors would like to thank H.L.S. Younes for valuable comments on a draft of this paper.

## References

[1] R. Albert and H. G. Othmer. The topology of the regulatory interactions predics the expression pattern of the segment polarity genes in drosophila melanogaster. *J. Theor. Biol.*, 223:1, 2003.

[2] M. Antoniotti, A. Policriti, N. Ugel, and B. Mishra. Model building and model checking for biochemical processes. *Cell Biochem Biophys.*, 38(3):271–286, 2003.

[3] C. Baier, E. M. Clarke, V. Hartonas-Garmhausen, M. Z. Kwiatkowska, and M. Ryan. Symbolic model checking for probabilistic processes. In *ICALP '97*, pages 430–440, London, UK, 1997. Springer-Verlag.

[4] 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.

[5] N.T.J. Bailey. *The Elements of Stochastic Processes with Applications to the Natural Sciences*. Wiley-IEEE, 1990.

[6] N. Barkai and S. Leibler. Biological rhythms: Circadian clocks limited by noise. *Nature*, 403:267–268, 2000.

[7] G. Batt, D. Ropers, H. de Jong, J. Geiselmann, R. Mateescu, M. Page, and D. Schneider. Validation of qualitative models of genetic regulatory networks by model checking: analysis of the nutritional stress response in Escherichia coli. *Bioinformatics*, 25(1):i19–i28, 2005.

[8] R.E. Bechhofer. A note on the limiting relative efficiency of the Wald sequential probability ratio test. *J. Amer. Statist. Assoc.*, 55:660–663, 1960.

[9] M. Calder, S. Gilmore, and J. Hillston. Modelling the influence of RKIP on the ERK signalling pathway using the stochastic process algebra PEPA. *Transactions on Computational Systems Biology*, page in press, 2006.

[10] M. Calder, V. Vyshemirsky, D. Gilbert, and R. Orton. Analysis of signalling pathways using the PRISM model checker. *Proc. Computational Methods in Systems Biology (CMSB'05)*, pages 179–190, 2005.

[11] L. Cardelli. Abstract machines of systems biology. *Comp. Sys. Biology*, 3737:145–168, 2005.

[12] N. Chabrier and F. Fages. Symbolic Model Checking of Biochemical Networks. *Proc 1st Internl Workshop on Computational Methods in Systems Biology*, pages 149–162, 2003.

[13] F. Ciesinski and M. Größer. On probabilistic computation tree logic. In *Validation of Stochastic Systems*, LNCS, 2925, pages 147–188. Springer, 2004.

[14] E. M. Clarke, O. Grumberg, and D. A. Peled. *Model Checking*. MIT Press, Cambridge, MA, 1999.

[15] 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.

[16] 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.

[17] Costas Courcoubetis and Mihalis Yannakakis. The complexity of probabilistic verification. *Journal of the ACM*, 42(4):857–907, 1995.

[18] P. Diaconis and D. Ylvisaker. Quantifying prior opinion. In J. M. Bernardo, M. H. De Groot, D. B. Lindley, and A. F. M. Smith, editors, *Bayesian Statistics 2: Proceedings of the 2nd Valencia International Meeting*. Elsevier Science Publisher, 1985.

[19] 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.

[20] Bernd Finkbeiner and Henny Sipma. Checking finite traces using alternating automata. In *In Proceedings of Runtime Verification (RV01) [1*, pages 44–60, 2001.

[21] J. Fisher, N. Piterman, E. J. Hubbard, M. J. Stern, and D. Harel. Computational insights into caenorhabditis elegans vulval development. *Proc Natl Acad Sci U S A*, 102(6):1951–1956, 2005.

[22] B.K. Ghosh and P.K. Sen, editors. *Handbook of sequential analysis*. Dekker, 1991.

[23] Daniel T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361, December 1977.

[24] R. Grosu and S.A. Smolka. Monte Carlo Model Checking. In *CAV*, pages 271–286, 2005.

[25] 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.

[26] 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.

[27] 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.

[28] A. Hinton, M. Kwiatkowska, G. Norman, and D. Parker. PRISM: A tool for automatic verification of probabilistic systems. In H. Hermanns and J. Palsberg, editors, *Proc. 12th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'06)*, volume 3920 of *LNCS*, pages 441–444. Springer, 2006.

[29] H. Jeffreys. *Theory of Probability*. Clarendon Press, Oxford, 1961.

[30] Naaman Kam, David Harel, and Irun R. Cohen. Modeling biological reactivity: Statecharts vs. boolean logic. In *In Proceeding of the Second International Conference on Systems Biology*, 2001.

[31] M. Z. Kwiatkowska, G. Norman, and D. Parker. Prism 2.0: A tool for probabilistic model checking. In *QEST*, pages 322–323. IEEE, 2004.

[32] P. Lecca and C. Priami. Cell cycle control in eukaryotes: A BioSpi model. In *Proc. Workshop on Concurrent Models in Molecular Biology (BioConcur'03)*, ENTCS, 2003.

[33] H. McAdams and L. Shapiro. Circuit simulation of genetic networks. *Science*, 269:650–656, 1995.

[34] Susan S. Owicki and Leslie Lamport. Proving liveness properties of concurrent programs. *ACM Trans. Program. Lang. Syst.*, 4(3):455–495, 1982.

[35] Amir Pnueli. The temporal logic of programs. In *FOCS*, pages 46–57. IEEE, 1977.

[36] Corrado Priami, Aviv Regev, Ehud Shapiro, and William Silverman. Application of a stochastic name-passing calculus to representation and simulation of molecular processes. *Inf. Process. Lett.*, 80(1):25–31, 2001.

[37] Avital Sadot, Jasmin Fisher, Dan Barak, Yishai Admanit, Michael J. Stern, E. Jane Albert Hubbard, and David Harel. Toward verified biological models. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 5(2):223–234, 2008.

[38] Koushik Sen, Mahesh Viswanathan, and Gul Agha. Statistical model checking of black-box probabilistic systems. In *CAV*, LNCS 3114, pages 202–215. Springer, 2004.

[39] Koushik Sen, Mahesh Viswanathan, and Gul Agha. On statistical model checking of stochastic systems. In *CAV*, LNCS 3576, pages 266–280, 2005.

[40] J. Vilar, H.-Y. Kueh, N. Barkai, and S. Leibler. Mechanisms of noise-resistance in genetic oscillators. *Proc. Nat Acad Sci USA*, 99(9):5988–5992, 2002.

[41] A. Wald. Sequential tests of statistical hypotheses. *Annals of Mathematical Statistics*, 16(2):117–186, 1945.

[42] Abraham Wald. *Sequential Analysis*. Dover Publications, June 2004.

[43] T. M. Yi, H. Kitano, and M. I. Simon. A quantitative characterization of the yeast heterotrimeric g protein cycle. *Proc Natl Acad Sci USA*, 100(19):10764–10769, 2003.

[44] Håkan L. S. Younes, Marta Z. Kwiatkowska, Gethin Norman, and David Parker. Numerical vs. statistical probabilistic model checking. *STTT*, 8(3):216–228, 2006.

[45] 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.

[46] 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.

# Appendices

## A Bounded Sampling of Bounded LTL

For Statistical Model Checking, BLTL formulas need to be checkable on simulations after a finite duration of the simulation, because the simulation cannot be continued indefinitely. Like the semantics of the unbounded linear temporal logic LTL [35], the semantics of BLTL in Section 2 is defined on infinite traces with divergence of time. In practice, simulations are only finite prefixes of infinite traces and cannot be extended uniquely to an infinite trace. In this section we prove the following lemma, which shows well-definedness of the BLTL semantics on finite system simulations and decidability of BLTL on simulation traces. These results are crucial to make sense of BLTL properties on traces that can be obtained by simulating systems in finite time.

**Lemma 1 (Bounded sampling theorem)** *The problem "$\sigma \models \phi$" is well-defined and can be checked for BLTL formulas $\phi$ and traces $\sigma$ based on only a* finite *prefix of $\sigma$ of bounded duration.*

For proving Lemma 1 we need to derive bounds on when to stop simulation. The duration bound for which we can show that the BLTL semantics is well-defined can be read off easily from the BLTL formula:

**Definition 3** *We define the* sampling bound $\#(\phi) \in \mathbb{Q}_{\geqslant 0}$ *of a BLTL formula $\phi$ inductively as the maximum nested sum of time bounds:*

$$\begin{aligned}
\#(x \sim v) &:= 0 \\
\#(\neg\phi_1) &:= \#(\phi_1) \\
\#(\phi_1 \vee \phi_2) &:= \max(\#(\phi_1), \#(\phi_2)) \\
\#(\phi_1 \wedge \phi_2) &:= \max(\#(\phi_1), \#(\phi_2)) \\
\#(\phi_1 \mathbf{U}^t \phi_2) &:= t + \max(\#(\phi_1), \#(\phi_2))
\end{aligned}$$

Unlike infinite traces, actual system simulations do not have infinite length but need to be finite. The following result shows for which duration the simulation can be stopped so that the BLTL property has a well-defined semantics and will not change its truth-value by continuing the simulation. We prove that the semantics of BLTL formulas $\phi$ is well-defined on finite prefixes of traces with a duration that is bounded by $\#(\phi)$.

**Lemma 2 (Well-definedness of BLTL on bounded simulation traces)** *Let $\phi$ be a BLTL formula, $k \in \mathbb{N}$. Then for any two infinite traces $\sigma = (s_0,t_0),(s_1,t_1),\ldots$ and $\tilde{\sigma} = (\tilde{s_0},\tilde{t_0}),(\tilde{s_1},\tilde{t_1}),\ldots$ with*

$$s_I = \tilde{s}_I \text{ and } t_I = \tilde{t}_I \text{ for all } I \in \mathbb{N} \text{ with } \sum_{0 \leqslant l < I} t_{k+l} \leqslant \#(\phi) \tag{6}$$

*we have that*

$$\sigma^k \models \phi \text{ iff } \tilde{\sigma}^k \models \phi \ .$$

**Proof 1** *The proof is by induction on the structure of the BLTL formula $\phi$. IH is short for induction hypothesis.*

1. *If $\phi$ is of the form $x \sim v$, then $\sigma^k \models x \sim v$ iff $\tilde{\sigma}^k \models x \sim v$, because $s_k = \tilde{s}_k$ by using* (6) *for $i = 0$.*

2. *If $\phi$ is of the form $\phi_1 \vee \phi_2$, then*

$$\begin{aligned}
&\sigma^k \models \phi_1 \vee \phi_2 \\
&\text{iff } \sigma^k \models \phi_1 \text{ or } \sigma^k \models \phi_2 \\
&\text{iff } \tilde{\sigma}^k \models \phi_1 \text{ or } \tilde{\sigma}^k \models \phi_2 \qquad \text{by IH as } \#(\phi_1 \vee \phi_2) \geqslant \#(\phi_1) \text{ and } \#(\phi_1 \vee \phi_2) \geqslant \#(\phi_2) \\
&\text{iff } \tilde{\sigma}^k \models \phi_1 \vee \phi_2
\end{aligned}$$

*The proof is similar for $\neg\phi_1$ and $\phi_1 \wedge \phi_2$.*

3. *If $\phi$ is of the form $\phi_1 \mathbf{U}^t \phi_2$, then $\sigma^k \models \phi_1 \mathbf{U}^t \phi_2$ iff for some $i \in \mathbb{N}$ the following conditions hold:*

   *(a)* $\sum_{0 \leqslant l < i} t_{k+l} \leqslant t$,

   *(b)* $\sigma^{k+i} \models \phi_2$, *and*

   *(c) for each* $0 \leqslant j < i$, $\sigma^{k+j} \models \phi_1$.

   *These conditions (a),(b),(c) are equivalent, respectively, to the following conditions (a′),(b′),(c′):*

   *(a′)* $\sum_{0 \leqslant l < i} \tilde{t}_{k+l} \leqslant t$, *because* $\#(\phi_1 \mathbf{U}^t \phi_2) \geqslant t$ *such that the durations of trace* $\sigma$ *and* $\tilde{\sigma}$ *are* $t_{k+l} = \tilde{t}_{k+l}$ *for each index* $l$ *with* $0 \leqslant l < i$ *by assumption* (6).

   *(b′)* $\tilde{\sigma}^{k+i} \models \phi_2$ *by induction hypothesis as follows: We know that the traces* $\sigma$ *and* $\tilde{\sigma}$ *match at* $k$ *for duration* $\#(\phi_1 \mathbf{U}^t \phi_2)$ *and need to show that the semantics of* $\phi_1 \mathbf{U}^t \phi_2$ *matches at* $k$. *By IH we know that* $\phi_2$ *has the same semantics at* $k+i$ *(that is* $\tilde{\sigma}^{k+i} \models \phi_2$ *iff* $\sigma^{k+i} \models \phi_2$*) provided that we can show that the traces* $\sigma$ *and* $\tilde{\sigma}$ *match at* $k+i$ *for duration* $\#(\phi_2)$. *For this, consider any* $I \in \mathbb{N}$ *with* $\sum_{0 \leqslant l < I} t_{k+i+l} \leqslant \#(\phi_2)$. *Then*

$$\#(\phi_2) \geqslant \sum_{0 \leqslant l < I} t_{k+i+l} = \sum_{0 \leqslant l < i+I} t_{k+l} - \sum_{0 \leqslant l < i} t_{k+l} \overset{(a)}{\geqslant} \sum_{0 \leqslant l < i+I} t_{k+l} - t$$

   *Thus*

$$\sum_{0 \leqslant l < i+I} t_{k+l} \leqslant t + \#(\phi_2) \leqslant t + \max(\#(\phi_1), \#(\phi_2)) = \#(\phi_1 \mathbf{U}^t \phi_2)$$

   *As* $I \in \mathbb{N}$ *was arbitrary, we conclude from this with assumption* (6) *that, indeed*

$$s_I = \tilde{s}_I \text{ and } t_I = \tilde{t}_I \text{ for all } I \in \mathbb{N} \text{ with } \sum_{0 \leqslant l < I} t_{k+i+l} \leqslant \#(\phi_2)$$

   *Thus the IH for* $\phi_2$ *yields the equivalence of* $\sigma^{k+i} \models \phi_2$ *and* $\tilde{\sigma}^{k+i} \models \phi_2$ *when using the equivalence of (a) and (a′).*

   *(c′) for each* $0 \leqslant j < i$, $\tilde{\sigma}^{k+j} \models \phi_1$. *The proof of equivalence to (c) is similar to that for (b′)*

*using $j < i$.*

*The existence of an $i \in \mathbb{N}$ for which these conditions hold is equivalent to $\tilde{\sigma}^k \models \phi_1 \mathbf{U}^t \phi_2$.*

As a consequence, for checking $\sigma \models \phi$ during Statistical Model Checking, we can stop simulation of sample $\sigma$ at duration $\#(\phi)$. By divergence of time, this happens after a finite number of simulation steps.

Now we prove that Lemma 1 holds using prefixes of traces according to the sampling bound $\#(\phi)$, which guarantees that finite simulations are sufficient for deciding $\phi$. In particular, checks for "$\sigma \models \phi$" terminate. We do not stop simulation prematurely, i.e., before "$\sigma \models \phi$" can be checked.

**Proof 2 (of Lemma 1)** *According to Lemma 2, the decision "$\sigma \models \phi$" is uniquely determined (and well-defined) by considering only a prefix of $\sigma$ of duration $\#(\phi) \in \mathbb{Q}_{\geqslant 0}$. By divergence of time, $\sigma$ reaches or exceeds this duration $\#(\phi)$ in some finite number of steps $n$. Let $\sigma' = (s_0,t_0),(s_1,t_1),\ldots,(s_n,t_n)$ denote a finite prefix of $\sigma$ of length $n$ such that $\sum_{0 \leqslant l < n} t_l \geqslant \#(\phi)$. Again by Lemma 2, the semantics of $\sigma' \models \phi$ is well-defined because any extension $\sigma''$ of $\sigma'$ satisfies $\sigma'' \models \phi$ if and only if $\sigma' \models \phi$ . Consequently the semantics of $\sigma' \models \phi$ coincides with the semantics of $\sigma \models \phi$. On the finite trace $\sigma'$, it is easy to see that BLTL is decidable by evaluating the atomic formulas $x \sim v$ at each state $s_i$ of the system simulation.*

# B Bayes Factor for General Priors

We show how to compute the Bayes factor when the prior density is a mixture of Beta densities. Most textbooks on Bayesian Statistics address the simple, non-mixture case, so here we report the general case for completeness.

Suppose $G$ is a density over $(0,1)$ defined as

$$G(u) \mathrel{\widehat{=}} \sum_{i=1}^{N} r_i \cdot g_i(u, \alpha_i, \beta_i)$$

where $N$ is a positive integer, the $g_i$'s are Beta densities (of possibly different parameters $\alpha_i, \beta_i$), and the $r_i$'s are positive reals summing up to 1. Our data $(x_1, \ldots, x_n)$ are assumed to be iid samples drawn from a Bernoulli distribution of unknown parameter $p$, so the probability of observing $d = (x_1, \ldots, x_n)$ is

$$f(d|p) = p^x (1-p)^{n-x}$$

where $x = \sum_{i=1}^{n} x_i$ is the number of successes in $(x_1, \ldots, x_n)$. Specializing (2) the Bayes factor at stage $n$ is:

$$\begin{aligned}
&\mathcal{B}_n \\
&= \\
&\frac{\int_{\theta}^{1} f(d|u) G(u)\, du}{\int_{0}^{\theta} f(d|u) G(u)\, du} \\
&= && \text{definition of } G \\
&\frac{\int_{\theta}^{1} f(d|u) \sum_{i=1}^{N} r_i g_i(u, \alpha_i, \beta_i)\, du}{\int_{0}^{\theta} f(d|u) \sum_{i=1}^{N} r_i g_i(u, \alpha_i, \beta_i)\, du} \\
&= && \text{linearity of integration} \\
&\frac{\sum_{i=1}^{N} r_i \int_{\theta}^{1} f(d|u) g_i(u, \alpha_i, \beta_i)\, du}{\sum_{i=1}^{N} r_i \int_{0}^{\theta} f(d|u) g_i(u, \alpha_i, \beta_i)\, du} \\
&= && \text{definition of } f \text{ and } g_i
\end{aligned}$$

$$\frac{\sum_{i=1}^{N} \frac{r_i}{B(\alpha_i,\beta_i)} \int_{\theta}^{1} u^x (1-u)^{n-x} u^{\alpha_i-1} (1-u)^{\beta_i-1}\, du}{\sum_{i=1}^{N} \frac{r_i}{B(\alpha_i,\beta_i)} \int_{0}^{\theta} u^x (1-u)^{n-x} u^{\alpha_i-1} (1-u)^{\beta_i-1}\, du}$$

= introduce $r_i'$

$$\frac{\sum_{i=1}^{N} r_i' \int_{\theta}^{1} u^x (1-u)^{n-x} u^{\alpha_i-1} (1-u)^{\beta_i-1}\, du}{\sum_{i=1}^{N} r_i' \int_{0}^{\theta} u^x (1-u)^{n-x} u^{\alpha_i-1} (1-u)^{\beta_i-1}\, du}$$

= algebra and split integral at numerator

$$\frac{\sum_{i=1}^{N} r_i' \left( \int_{0}^{1} u^{x+\alpha_i-1} (1-u)^{n-x+\beta_i-1}\, du - \int_{0}^{\theta} u^{x+\alpha_i-1} (1-u)^{n-x+\beta_i-1}\, du \right)}{\sum_{i=1}^{N} r_i' \int_{0}^{\theta} u^{x+\alpha_i-1} (1-u)^{n-x+\beta_i-1}\, du}$$

= split fraction and simplify

$$\frac{\sum_{i=1}^{N} r_i' \int_{0}^{1} u^{x+\alpha_i-1} (1-u)^{n-x+\beta_i-1}\, du}{\sum_{i=1}^{N} r_i' \int_{0}^{\theta} u^{x+\alpha_i-1} (1-u)^{n-x+\beta_i-1}\, du} - 1$$

= definition of Beta function (4)

$$\frac{\sum_{i=1}^{N} r_i' B(x+\alpha_i, n-x+\beta_i)}{\sum_{i=1}^{N} r_i' \int_{0}^{\theta} u^{x+\alpha_i-1} (1-u)^{n-x+\beta_i-1}\, du} - 1$$

= definition of Beta distribution function (5)

$$\frac{\sum_{i=1}^{N} r_i' B(x+\alpha_i, n-x+\beta_i)}{\sum_{i=1}^{N} r_i' B(x+\alpha_i, n-x+\beta_i) F_{(x+\alpha_i, n-x+\beta_i)}(\theta)} - 1\,.$$

where $r'_i = \frac{r_i}{B(\alpha_i,\beta_i)}$. For the special case $N = 1$ the Bayes factor at stage $n$ is simply

$$\mathcal{B}_n = \frac{1}{F_{(x+\alpha,n-x+\beta)}(\theta)} - 1 \ .$$

# C Termination of Bayesian Model Checking Algorithm

## C.1 Termination for Beta priors

The Beta distribution of real parameters $\alpha, \beta > 0$ is defined on $(0,1)$ by the density

$$g(u,\alpha,\beta) \mathrel{\widehat{=}} \frac{1}{B(\alpha,\beta)} u^{\alpha-1}(1-u)^{\beta-1}$$

where $B(\alpha,\beta) \mathrel{\widehat{=}} \int_0^1 t^{\alpha-1}(1-t)^{\beta-1}dt$.

We shall later need the following facts about binomial expansions. For positive integer $n$ and real $\theta$ it is well known that:

$$(1-\theta)^n = \sum_{i=0}^{n} \binom{n}{i} (-1)^i \theta^i \ .$$

The above result can be generalized to an arbitrary real $r$ for $\theta \in (-1,1)$:

$$(1-\theta)^r = \sum_{i=0}^{\infty} \binom{r}{i} (-1)^i \theta^i \tag{7}$$

where

$$\binom{r}{i} = \frac{r(r-1)\cdots(r-i+1)}{i!} \ .$$

For the special case $r > -1$ and $\theta = -1$ we have that:

$$2^r = \sum_{i=0}^{\infty} \binom{r}{i} \ . \tag{8}$$

Since $\left|\binom{r}{i}\theta^i\right| \leqslant \binom{r}{i}$ for $\theta \in (-1,1)$ and the series (8) converges, by Weierstrass's criterion we deduce uniform convergence of (7) for $\theta \in (-1,1)$. This implies that when integrating the binomial series - as we shall later need - one can interchange the operation of limit sum and integration.

**Proof 3 (Theorem 1)** *Suppose $X$ is a Bernoulli random variable of (unknown) parameter $p$. The algorithm iteratively and independently draws samples of $X$ (denoted by $x_i$ for $i \in \mathbb{N}$). The random variables $X_i$ corresponding to the $x_i$ are thus independent and identically distributed (iid). From Definition 2, the Bayes factor $\mathcal{B}_n$ at stage $n$ is:*

$$\mathcal{B}_n \widehat{=} \frac{P(X_1,\ldots,X_n|H_0)}{P(X_1,\ldots,X_n|H_1)} .$$

*Given an arbitrary threshold $T > 1$, the algorithm stops at stage $n$ iff $(\mathcal{B}_n > T \ \vee \ \mathcal{B}_n < \frac{1}{T})$. We show that this happens with probability one.*

*Our data $x_i$ are assumed to be iid samples drawn from a Bernoulli distribution of unknown parameter $p$, so the probability of observing $d = (x_1,\ldots,x_n)$ is*

$$f(d|p) = p^x(1-p)^{n-x}$$

*where $x$ is the number of successes in $(x_1,\ldots,x_n)$. The hypotheses to test are $H_0 : p \geqslant \theta$ vs. $H_1 : p < \theta$, where $\theta$ is a fixed real in $(0,1)$ from the PBLTL property. The prior density $g(\cdot)$ is assumed to be a Beta density with fixed parameters $\alpha,\beta > 0$. Specializing (2) the Bayes factor at stage $n$ is thus:*

$$\begin{aligned} \mathcal{B}_n &= \frac{\int_\theta^1 f(d|u)g(u)\,du}{\int_0^\theta f(d|u)g(u)\,du} = \frac{\frac{1}{B(\alpha,\beta)}\int_\theta^1 u^x(1-u)^{n-x}u^{\alpha-1}(1-u)^{\beta-1}\,du}{\frac{1}{B(\alpha,\beta)}\int_0^\theta u^x(1-u)^{n-x}u^{\alpha-1}(1-u)^{\beta-1}\,du} \\ &= \frac{\int_\theta^1 u^x(1-u)^{n-x}u^{\alpha-1}(1-u)^{\beta-1}\,du}{\int_0^\theta u^x(1-u)^{n-x}u^{\alpha-1}(1-u)^{\beta-1}\,du} = \frac{I(\theta,1)}{I(0,\theta)} \end{aligned} \tag{9}$$

*where $I(a,b)$ is*

$$I(a,b) \widehat{=} \int_a^b u^{x+\alpha-1}(1-u)^{n-x+\beta-1}\, du\ .$$

*We now simplify the integral term, and have that*

$$\begin{aligned}
& I(a,b) \\
&= && \textit{binomial expansion (7)} \\
& \int_a^b u^{x+\alpha-1} \sum_{i=0}^{\infty} \binom{n-x+\beta-1}{i} (-1)^i u^i\, du \\
&= && \textit{uniform convergence} \\
& \sum_{i=0}^{\infty} \binom{n-x+\beta-1}{i} \int_a^b (-1)^i u^{i+x+\alpha-1}\, du \\
&= && \textit{solve integral} \\
& \sum_{i=0}^{\infty} \binom{n-x+\beta-1}{i} \frac{(-1)^i}{i+x+\alpha}\, u^{i+x+\alpha}\Big|_a^b \\
&= && \textit{notation } c_i \widehat{=} \tbinom{n-x+\beta-1}{i} \tfrac{(-1)^i}{i+x+\alpha} \\
& \sum_{i=0}^{\infty} c_i\, u^{i+x+\alpha}\Big|_a^b \\
&= && \textit{expand primitive} \\
& \sum_{i=0}^{\infty} c_i\, (b^{i+x+\alpha} - a^{i+x+\alpha})
\end{aligned}$$

*and we now introduce the notation $S(a,b)$ for the sum above*

$$S(a,b) \widehat{=} \sum_{i=0}^{\infty} c_i\, (b^{i+x+\alpha} - a^{i+x+\alpha}) = I(a,b) \tag{10}$$

*where $c_i \widehat{=} \binom{n-x+\beta-1}{i} \frac{(-1)^i}{i+x+\alpha}$ (we recall that n is the number of samples and x the number of successes). Since $P(X_1,\ldots,X_n \,|\, a<p<b) = \frac{S(a,b)}{B(\alpha,\beta)}$, we have that $S(a,b)$ must be strictly positive for*

*any $a < b$ in $[0,1]$, that is:*

$$\forall n \ \forall x \leqslant n \ \ \forall 0 \leqslant a < b \leqslant 1 \qquad \sum_{i=0}^{\infty} c_i \ (b^{i+x+\alpha} - a^{i+x+\alpha}) > 0 \ . \tag{11}$$

*Finally, our aim is to establish whether the algorithm stops at stage $n$* i.e.*, whether for some $n$ it is true that $(\mathcal{B}_n > T \ \vee \ \mathcal{B}_n < \frac{1}{T})$. A sufficient condition for termination is to show that the algorithm accepts $H_0$ with probability one, unless it has already rejected $H_0$ (where the algorithm terminates anyhow, accepting $H_1$). We therefore consider the likelihood that $\mathcal{B}_n > T$ becomes true when $\mathcal{B}_i \geqslant \frac{1}{T}$ for $0 \leqslant i < n$.*

*From the definition of S, we can see that $S(a,b)$ is an integral from a to b. By (9), we can rewrite $\mathcal{B}_n$:*

$$\mathcal{B}_n = \frac{S(\theta,1)}{S(0,\theta)} = \frac{S(0,1) - S(0,\theta)}{S(0,\theta)} = \frac{S(0,1)}{S(0,\theta)} - 1 \ . \tag{12}$$

*We reason:*

$\mathcal{B}_n > T$

$\equiv$ *(12) and $S(0,\theta)$ positive*

$S(0,1) - (T+1)S(0,\theta) > 0$

$\equiv$ *definition of S*

$\sum_{i=0}^{\infty} c_i \ (1 - (T+1)\theta^{i+x+\alpha}) > 0$

$\equiv$ *algebra using $(T+1) > 0$*

$\sum_{i=0}^{\infty} c_i \ (1 - ((T+1)^{\frac{1}{i+x+\alpha}}\theta)^{i+x+\alpha}) > 0$

*Using (11) for $b = 1$ we know that $\sum_{i=0}^{\infty} c_i \ (1 - a^{i+x+\alpha}) > 0$ for all $0 \leqslant a < 1$. Therefore, a sufficient condition to make $\mathcal{B}_n > T$ true is to make $(T+1)^{\frac{1}{i+x+\alpha}}\theta < 1$. That amounts to find an x such that*

*for all i*

$$(T+1)^{\frac{1}{i+x+\alpha}}\theta < 1$$

$\equiv$ *apply logarithm*

$$\frac{1}{i+x+\alpha}\log(T+1)+\log\theta < 0$$

$\equiv$ $\alpha > 0, x \geqslant 0, i \geqslant 0$ *and algebra*

$$\log(T+1) < -(i+x+\alpha)\log\theta$$

$\equiv$ *property of logarithm*

$$\log(T+1) < (i+x+\alpha)\log\tfrac{1}{\theta}$$

$\equiv$ $0 < \theta < 1$

$$\frac{\log(T+1)}{\log\frac{1}{\theta}} < (i+x+\alpha)$$

*which will be eventually true with probability one, as long as the unknown probability of success p is non-zero. (Note that it is sufficient to consider the case* $i = 0$*.) We thus have to prove that the event* $\bigcup_{n=1}^{\infty}(k < x_{n,p})$ *has probability 1, where x is distributed as a binomial of parameters n and* $p > 0$*, and* $k = \lceil \frac{\log(T+1)}{\log\frac{1}{\theta}} - \alpha \rceil$*. We reason:*

$$P(\bigcup_{n=1}^{\infty}(k < x_{n,p}))$$

$=$ *probability measures are continuous*

$$\lim_{n\to\infty} P(k < x_{n,p})$$

$=$ *complemented event*

$$\lim_{n\to\infty} 1 - P(x_{n,p} \leqslant k)$$

$=$ *disjoint events*

$$1 - \lim_{n\to\infty} \sum_{i=0}^{k} P(x_{n,p} = i)$$

$=$ *x distributed as binomial of parameters* $n, p$

$1 - \lim_{n\to\infty} \sum_{i=0}^{k} \binom{n}{i} p^i (1-p)^{n-i}$

$=$ *continuity of finite sums (assume* $0 < p < 1$*)*

$1 - \sum_{i=0}^{k} (\frac{p}{1-p})^i \lim_{n\to\infty} (1-p)^n \binom{n}{i}$

$=$ *expand binomial coefficient*

$1 - \sum_{i=0}^{k} (\frac{p}{1-p})^i \frac{1}{i!} \lim_{n\to\infty} (1-p)^n n(n-1)\cdots(n-i+1)$

$=$ $0 < p < 1$ *and limit*

$1 - \sum_{i=0}^{k} 0 = 1$ .

*The case* $p = 1$ *follows directly from the third to last step. For* $p = 0$*, instead, we have* $x = 0$ *for any number of samples n, so that it is easy to see from (9) that* $\mathcal{B}_n \to 0$ *for* $n \to \infty$*, and* $H_1$ *will be accepted eventually. In fact:*

$$\mathcal{B}_n = \frac{\displaystyle\int_\theta^1 (1-u)^n u^{\alpha-1}(1-u)^{\beta-1}\,du}{\displaystyle\int_0^\theta (1-u)^n u^{\alpha-1}(1-u)^{\beta-1}\,du} \leqslant \frac{\displaystyle\int_\theta^1 u^{\alpha-1}(1-u)^{\beta-1}\,du}{(1-\theta)^n \displaystyle\int_0^\theta u^{\alpha-1}(1-u)^{\beta-1}\,du}$$

*and since* $0 < \theta < 1$ *we therefore have* $\mathcal{B}_n \to 0$ *for* $n \to \infty$*.*

## C.2 Termination for General Priors

Suppose $G$ is a density over $(0,1)$ defined as

$$G(u) \mathrel{\widehat{=}} \sum_{j=1}^{N} r_j \cdot g_j(u, \alpha_j, \beta_j)$$

where $N$ is a positive integer, the $g_j$'s are Beta densities (of possibly different parameters $\alpha_j, \beta_j$), and the $r_j$'s are positive reals summing up to 1. We want to show that our algorithm terminates with probability one when $G$ is used as a prior. We shall retain much of the notation and concepts already introduced.

From the derivation in Appendix B we have that the Bayes factor $\mathcal{B}_n$ at stage $n$ is

$$\mathcal{B}_n = \frac{\sum_{j=1}^{N} r'_j \int_0^1 u^{x+\alpha_j-1}(1-u)^{n-x+\beta_j-1}\,du}{\sum_{j=1}^{N} r'_j \int_0^\theta u^{x+\alpha_j-1}(1-u)^{n-x+\beta_j-1}\,du} - 1 = \frac{\sum_{j=1}^{N} r'_j I_j(0,1)}{\sum_{j=1}^{N} r'_j I_j(0,\theta)} - 1 \qquad (13)$$

where $r'_j = \frac{r_j}{B(\alpha_i,\beta_i)}$ and $I_j(a,b)$ is a slight generalization of $I(a,b)$:

$$I_j(a,b) \mathrel{\widehat{=}} \int_a^b u^{x+\alpha_j-1}(1-u)^{n-x+\beta_j-1}\,du\ .$$

In analogy to what we proved in Appendix C.1, we show that the algorithm accepts $H_0$ with probability one, unless it has already rejected it before. We thus have to show that $\mathcal{B}_n > T$ with probability one, when $\mathcal{B}_i \geqslant \frac{1}{T}$ for $i < n$. The strategy we use is first to find an expression $\mathcal{B}'_n$ such that for all $n$ $\mathcal{B}'_n \leqslant \mathcal{B}_n$. Then, we prove that with probability one there is a $z$ such that $\mathcal{B}'_z > T$, which in turn implies $\mathcal{B}_z > T$ and termination of the algorithm (accepting $H_0$) with probability one.

We now reason from (13):

$$\begin{array}{ll} \mathcal{B}_n & \\ \geqslant & R = \max_j r'_j \\ \dfrac{\sum_{j=1}^{N} r'_j I_j(0,1)}{R\sum_{j=1}^{N} I_j(0,\theta)} - 1 & \end{array}$$

$=$ definition of $I_j$

$$\frac{\sum_{j=1}^{N} r'_j I_j(0,1)}{R\sum_{j=1}^{N}\int_0^\theta u^{x+\alpha_j-1}(1-u)^{n-x+\beta_j-1}\,du} - 1$$

$=$ linearity of integration, laws of powers

$$\frac{\sum_{j=1}^{N} r'_j I_j(0,1)}{R\int_0^\theta u^{x-1}(1-u)^{n-x-1}\left(\sum_{j=1}^{N} u^{\alpha_j}(1-u)^{\beta_j}\right)\,du} - 1$$

$\geqslant$ $A = \min_j \alpha_j$ and $B = \min_j \beta_j$, monotonicity of integration

$$\frac{\sum_{j=1}^{N} r'_j I_j(0,1)}{R\int_0^\theta u^{x+A-1}(1-u)^{n-x+B-1} N\,du} - 1$$

$=$ algebra

$$\sum_{j=1}^{N} \frac{r'_j}{RN} \frac{I_j(0,1)}{\int_0^\theta u^{x+A-1}(1-u)^{n-x+B-1}\,du} - 1$$

and we have thus established that

$$\forall n \quad \mathcal{B}'_n \mathrel{\widehat{=}} \sum_{j=1}^{N} \frac{r'_j}{RN} \frac{I_j(0,1)}{\int_0^\theta u^{x+A-1}(1-u)^{n-x+B-1}\,du} - 1 \leqslant \mathcal{B}_n\,. \tag{14}$$

Now, to prove that eventually $\mathcal{B}_n > T$ we show that $\mathcal{B}'_n > T$. In particular, we show that one particular summand of $\mathcal{B}'_n$ can grow arbitrarily large (with probability one). Then, by the fact that the summands of $\mathcal{B}'_n$ are positive and by (14), we shall conclude $\mathcal{B}_n > T$ and termination of the algorithm (accepting $H_0$).

To prove $\mathcal{B}'_n > T$ it is thus sufficient to show that, with probability one, there are naturals $n$ and $x \leqslant n$ such that

$$\frac{I_k(0,1)}{\int_0^\theta u^{x+A-1}(1-u)^{n-x+B-1}\,du} > T \tag{15}$$

where $k$ is such that $\beta_k = B$. By the reasoning for $I(a,b)$ set out in Appendix C.1, we can rewrite (15):

$$\frac{\int_0^1 u^{x+\alpha_k-1}(1-u)^{n-x+B-1}\,du}{\int_0^\theta u^{x+A-1}(1-u)^{n-x+B-1}\,du} > T$$

$\equiv$ definition of $S$ (10)

$$\frac{\sum_{i=0}^{\infty}\binom{n-x+B-1}{i}\frac{(-1)^i}{i+x+\alpha_k}}{\sum_{i=0}^{\infty}\binom{n-x+B-1}{i}\frac{(-1)^i}{i+x+A}\theta^{i+x+A}} > T$$

$\equiv$ notation $c_i = \binom{n-x+B-1}{i}\frac{(-1)^i}{i+x+\alpha_k}$ and $\alpha_k > 0$

$$\frac{\sum_{i=0}^{\infty} c_i}{\sum_{i=0}^{\infty} c_i \frac{i+x+\alpha_k}{i+x+A}\theta^{i+x+A}} > T$$

$\equiv$ introduce $\theta^{\alpha_k}$

$$\frac{\sum_{i=0}^{\infty} c_i}{\sum_{i=0}^{\infty} c_i \frac{i+x+\alpha_k}{i+x+A}\theta^{A-\alpha_k}\,\theta^{i+x+\alpha_k}} > T$$

$\equiv$ positive denominator

$$\sum_{i=0}^{\infty} c_i > \sum_{i=0}^{\infty} c_i \frac{i+x+\alpha_k}{i+x+A} T\theta^{A-\alpha_k}\,\theta^{i+x+\alpha_k}$$

$\equiv$ algebra

$$\sum_{i=0}^{\infty} c_i \left(1 - \frac{i+x+\alpha_k}{i+x+A} T\theta^{A-\alpha_k}\ \theta^{i+x+\alpha_k}\right) > 0$$

$\equiv$ laws of powers

$$\sum_{i=0}^{\infty} c_i \left(1 - \left(\left(\frac{i+x+\alpha_k}{i+x+A} T\theta^{A-\alpha_k}\right)^{\frac{1}{i+x+\alpha_k}} \theta\right)^{i+x+\alpha_k}\right) > 0$$

For $b = 1$ in (11) we get that $\sum_{i=0}^{\infty} c_i\ (1 - a^{i+x+\alpha_k}) > 0$ for all $0 \leqslant a < 1$. Therefore, a sufficient condition to make (15) true is to find $x$ such that

$(\frac{i+x+\alpha_k}{i+x+A} T\theta^{A-\alpha_k})^{\frac{1}{i+x+\alpha_k}} \theta < 1$

$\equiv$ apply logarithm

$\frac{1}{i+x+\alpha_k} \log(\frac{i+x+\alpha_k}{i+x+A} T\theta^{A-\alpha_k}) + \log\theta < 0$

$\Leftarrow$ $\frac{i+x+\alpha_k}{i+x+A} \leqslant \frac{\alpha_k}{A}$, log monotonicity

$\frac{1}{i+x+\alpha_k} \log(\frac{\alpha_k}{A} T\theta^{A-\alpha_k}) + \log\theta < 0$

$\equiv$ algebra and $i+x+\alpha_k > 0$

$\log(\frac{\alpha_k}{A} T\theta^{A-\alpha_k}) < -(i+x+\alpha_k)\log\theta$

$\equiv$ law of logarithms

$\log(\frac{\alpha_k}{A} T\theta^{A-\alpha_k}) < (i+x+\alpha_k)\log\frac{1}{\theta}$

$\equiv$ $0 < \theta < 1$, thus $\log\theta < 0$

$$\frac{\log(\frac{\alpha_k}{A} T\theta^{A-\alpha_k})}{\log\frac{1}{\theta}} < (i+x+\alpha_k)$$

which is true with probability one, as we have already proven ($x$ grows arbitrarily large with probability one, when $p > 0$). Again, for the case $p = 0$ it is easy to see from (13) that $\mathcal{B}_n \to 0$ as $n \to \infty$, so that the algorithm eventually terminates by rejecting $H_0$.

# D Error Analysis

**Proof 4** *Suppose the algorithm terminates after observing n samples. Let X be the random variable denoting the number of observed traces satisfying the BLTL formula. Also, the probability with which traces from the model actually satisfy the BLTL formula is given by p, where* $p \geqslant \theta$. *The notation* $x \bowtie D$ *is used to indicate the event that x is drawn from the probability distribution D.*

*Also, we know that the expression* $t^i(1-t)^{n-i}$ *defined on* $[\theta, 1]$ *assumes a maximum value as it is a continuous function on a compact set. In particular, the maximum value is obtained when either* $t = \theta$ *or* $t = \frac{2i}{n}$. *We call this value of t as* $t_{max}$.

$$P(\textit{Type I error})$$

$$= \qquad \textit{By Definition}$$

$$P(H_1 \textit{ is chosen} \mid H_0 \textit{ is true})$$

$$= \qquad \textit{Since, } H_1 \textit{ is chosen iff } \mathcal{B}(n,X) < \tfrac{1}{T}$$

$$P\left(\{\mathcal{B}(n,X) < \frac{1}{T} \textit{ and } X \bowtie \textit{Binomial}(n,p)\} \mid H_0 \textit{ is true}\right)$$

$$= \qquad \textit{By Definition of Null Hypothesis and } p$$

$$P\left(\{\mathcal{B}(n,X) < \frac{1}{T} \textit{ and } X \bowtie \textit{Binomial}(n,p)\} \mid p \geqslant \theta\right)$$

$$= \qquad \textit{Definition of Conditional Probability}$$

$$\frac{P\left(\{\mathcal{B}(n,X) < \frac{1}{T} \textit{ and } X \bowtie \textit{Binomial}(n,p) \textit{ and } p \geqslant \theta\}\right)}{P(p \geqslant \theta)}$$

$$= \qquad \textit{By Definition (X can take values from 0 to n)}$$

$$\frac{P\left(\bigcup_{x=0}^{n} \{\mathcal{B}(n,X=x) < \frac{1}{T} \textit{ and } x \bowtie \textit{Binomial}(n,p) \textit{ and } p \geqslant \theta\}\right)}{P(p \geqslant \theta)}$$

$=$ *Disjoint Events*

$$\frac{\sum_{x=0}^{n} P\left(\{\mathcal{B}(n,x) < \frac{1}{T} \text{ and } x \bowtie Binomial(n,p) \text{ and } p \geqslant \theta\}\right)}{P(p \geqslant \theta)}$$

$=$ *Independence of Events*

$$\frac{\sum_{x=0}^{n} P\left(\{\mathcal{B}(n,x) < \frac{1}{T}\}\right) P\left(\{x \bowtie Binomial(n,p) \text{ and } p \geqslant \theta\}\right)}{P(p \geqslant \theta)}$$

$=$ *Algebraic Manipulation*

$$\sum_{x=0}^{n} P(\{\mathcal{B}(n,x) < \frac{1}{T}\}) \frac{P(\{x \bowtie Binomial(n,p) \text{ and } p \geqslant \theta\})}{P(p \geqslant \theta)}$$

$=$ *Definition of Conditional Probability*

$$\sum_{x=0}^{n} P(\{\mathcal{B}(n,x) < \frac{1}{T}\}) P\left(\{x \bowtie Binomial(n,p) \mid p \geqslant \theta\}\right)$$

$=$ *Conditional Probability*

$$\sum_{x=0}^{n} P(\{\ \mathcal{B}(n,x) < \frac{1}{T}\}) \binom{n}{x} p^x (1-p)^{n-x}, \text{ where } p \geqslant \theta$$

$=$ *I is indicator function*[3]

$$\sum_{x=0}^{n} I_{\{\mathcal{B}(n,x) < 1/T\}}(x) \binom{n}{x} p^x (1-p)^{n-x}, \text{ where } p \geqslant \theta$$

$\leqslant$ *Since, $t_{max}$ maximizes $p^x(1-p)^{n-x}$*

$$\sum_{x=0}^{n} I_{\{\mathcal{B}(n,x) < 1/T\}}(x) \binom{n}{x} t_{max}^x (1-t_{max})^{n-x}$$

3. $P(x \in A)$ is usually rewritten as $I_A(x)$ if $x \in A$ is known with probability 1 when $x$ and $A$ are known.