
Discovering rare behaviours in stochastic differential equations using decision procedures: applications to a minimal cell cycle model

Arup Kumar Ghosh* and Faraz Hussain

Computer Science Department,
University of Central Florida,
Orlando FL 32816, USA
Email: akghosh@cs.ucf.edu
Email: fhussain@cs.ucf.edu
*Corresponding author

Susmit Jha

Strategic CAD Labs,
Intel, Portland, OR 97124, USA
Email: susmit.jha@intel.com

Christopher J. Langmead

Computer Science Department,
Carnegie Mellon University,
Pittsburgh, PA 15213, USA
and
Lane Center for Computational Biology,
Carnegie Mellon University,
Pittsburgh PA 15213, USA
Email: cjl@cs.cmu.edu

Sumit Kumar Jha

Computer Science Department,
University of Central Florida,
Orlando FL 32816, USA
Email: jha@cs.ucf.edu

Abstract: Stochastic Differential Equation (SDE) models are used to describe the dynamics of complex systems with inherent randomness. The primary purpose of these models is to study rare but interesting or important behaviours, such as the formation of a tumour. Stochastic simulations are the most common means for estimating (or bounding) the probability of rare behaviours, but the cost of simulations increases with the rarity of events. To address this problem, we introduce a new algorithm specifically designed to quantify the likelihood of rare behaviours in SDE models. Our approach relies on temporal logics for specifying rare behaviours of interest, and on the ability of bit-vector decision procedures to reason exhaustively about fixed-precision

arithmetic. We apply our algorithm to a minimal parameterised model of the cell cycle, and take Brownian noise into account while investigating the likelihood of irregularities in cell size and time between cell divisions.

Keywords: bioinformatics; computational systems biology; rare behaviours; SDE; stochastic differential equation; decision procedures; cell cycle model; bit-vector; SMT solver; satisfiability modulo theories; stochastic model; Brownian noise; temporal logic.

Reference to this paper should be made as follows: Ghosh, A.K., Hussain, F., Jha, S., Langmead, C.J. and Jha, S.K. (2014) ‘Discovering rare behaviours in stochastic differential equations using decision procedures: applications to a minimal cell cycle model’, *Int. J. Bioinformatics Research and Applications*, Vol. 10, Nos. 4/5, pp.540–558.

Biographical notes: Arup Kumar Ghosh is a PhD candidate at the Department of Computer Science of University of Central Florida. His research focuses on algorithms, big data, discovery and analysis of rare behaviours in stochastic models, automated verification and synthesis of stochastic and hybrid systems using high level behavioural specifications. He received his MS in Computer Engineering from the University of Central Florida.

Faraz Hussain is a doctoral candidate in the EECS department at the University of Central Florida. His primary research interest is in developing formal methods techniques for analysing probabilistic computational models.

Susmit Jha is a Research Scientist in Strategic CAD Lab at Intel, Portland, OR. His research focuses on the automated synthesis of programs and cyber-physical systems from high level correctness, security and performance specifications.

Christopher J. Langmead is an Associate Professor in the School of Computer Science at Carnegie Mellon University. He has been on the Carnegie Mellon faculty since 2004.

Sumit Kumar Jha is the Charles N. Millican Assistant Professor of Computer Science at the University of Central Florida. His research interests include formal methods, high-performance computing, algorithms, and stochastic systems with applications in computational systems biology, cyber-physical systems, and computational finance.

This paper is a revised and expanded version of a paper entitled ‘Decision procedure based discovery of rare behaviors in stochastic differential equation models of biological systems’ presented at the ‘2nd IEEE International Conference on Computational Advances in Bio and Medical Sciences (ICCABS)’, Las Vegas, NV, USA, 23–25 February 2012.

1 Introduction

Complex biological systems whose behaviour depends on one or more sources of randomness are often described using Stochastic Differential Equation (SDE) models (Arato, 2003; Pfeuty and Kaneko, 2007; Ghosh et al., 2012; Jha and Langmead, 2011a; Lefever and Garay, 1978). SDEs are well suited to the investigation of biological

phenomena (e.g. bi-stability in genetic circuits; de Jong, 2002) and cyber-physical systems that are sensitive to stochastic effects (Marwedel, 2008). Unfortunately, many SDE models can be difficult to analyse algorithmically due to their stochastic nature. In particular, SDE models generally do not admit analytic solutions (except for very restricted forms). For this reason, stochastic simulations are generally used to understand the behaviour of such models. To perform such simulations, independent and identically distributed (iid) simulation trajectories are sampled from the model. Each sample can then be evaluated with respect to some *user-specified* behaviour (e.g. whether a tumour forms or a biological cyber-physical system crashes). The statistics of the resulting Bernoulli observations can then be used to estimate the probability with which the behaviour holds under the model and to estimate conditional probabilities (e.g. the probability that the tumour is caused by mutations to a particular cell type).

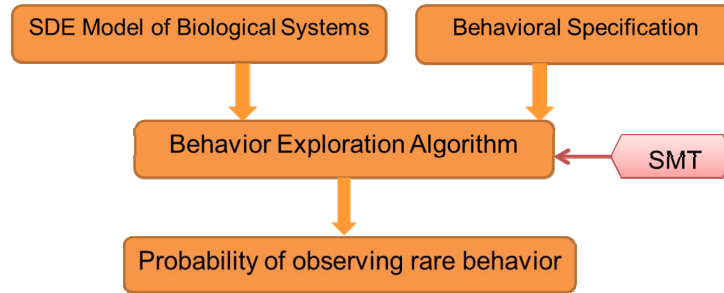
Independent and identically distributed-sampling is efficient if the behaviour under consideration is common, but many important behaviours are rare (Juneja and Shahabuddin, 2006). Sampling is an inefficient means for studying rare behaviours because the vast majority of the sampled trajectories will not exhibit the desired behaviour. For example, consider an SDE model of a population of cells. If the model is realistic, then phenomena such as tumour formation will be rare, as they are in nature. Consequently, the cost of generating multiple tumour-forming trajectories for subsequent analysis using iid sampling will become prohibitive. One natural approach for dealing with the cost of studying rare behaviours is to employ biased (i.e. non-iid) sampling procedures (e.g. Jha and Langmead, 2011a). In this approach, the underlying probability distribution over trajectories is manipulated in some fashion to expose rare behaviours. We note, however, that biased sampling schemes are usually insensitive to the details of the particular behaviour under consideration and may not achieve the desired goal of studying rare behaviours. Indeed, a poorly designed biased sampling scheme can actually increase the costs of studying rare behaviours.

A related challenge associated with SDEs (and stochastic models in general) is the need to ‘debug’ models during development. A model is flawed if it cannot exhibit known behaviours. Clearly, sampling cannot certify the non-existence of a rare behaviour.

On the other hand, decision procedures including Satisfiability Modulo Theories (SMT; Jha et al., 2009) are capable of producing proofs of infeasibility, even for classes of continuous dynamical systems (Jha et al., 2007). The algorithm presented in this paper employs decision procedures to address the problem of studying rare behaviours and debugging the associated stochastic model. Figure 1 gives an overview of our proposed approach.

Our algorithm can analyse SDE models of complex systems for *a priori* known behaviour. The known behaviour can be specified using formal specifications including probabilistic flavours of temporal logic. The algorithm then uses SMT-based decision procedures to explore all possible behaviours of the stochastic biological models without explicitly enumerating them. If no discrete behaviour with a probability density higher than a given threshold is found, the algorithm reports that the model is incapable of showing the stated behaviour with the given probability density. Otherwise, our algorithm produces a discrete behaviour of the model conforming to the given specification.

Figure 1 Exploring rare behaviours of SDE models of biological systems: an overview of our proposed approach. Our algorithm analyses SDE models of complex systems for *a priori* known behaviour. It takes the SDE model and behavioural specifications as inputs and explores rare behaviours by using satisfiability modulo theory-based decision procedures (see online version for colours)



2 Related work

We are inspired by developments in the fields of formal methods and software verification and have exploited the tools and techniques developed by researchers in these areas to better studying rare behaviours in stochastic systems. Our approach draws on concepts and ideas from the theory and practice of formal verification of computer hardware and software, such as model checking (Baier and Katoen, 2008). The vast majority of these model checking methods are intended for either finite-state models or finite-state abstractions of infinite state systems. Differential equation models, naturally, are defined on continuous domains. While there are a handful of formal methods suitable for analysing Ordinary Differential Equation (ODEs) (e.g. Piazza et al., 2005) and stochastic Continuous Time Markov Chain (CTMC) models (Clarke et al., 2008; Heath et al., 2006; Jha et al., 2009), none are suitable for SDE models, with the exception of our previous work (Jha and Langmead, 2011a) which uses a combination of non-iid sampling, Bayesian statistical hypothesis testing and Girsanov's theorem for change of measures to bound the probability of rare events. As the sampling procedure is not guided by the behaviour that is trying to discover, it is not surprising that our previous approach has limitations. In particular, the approach can never demonstrate that a model is incapable of demonstrating a specific behaviour with a given certainty.

The key difference between our earlier work (Jha and Langmead, 2011a) and the research presented in this paper is that we employ decision procedures to guide the sampling, whereas our previous paper used change of measures. Decision procedures based on SMT have been previously used for the formal verification of the correctness of software and hardware models (Gulwani et al., 2011; Cohen et al., 2009; Jha et al., 2010a), but not for complex stochastic models. Traditional research in SMT solving technique has focused on linear constraints with simple arithmetic and rich Boolean structure. However, the analysis of continuous and probabilistic dynamical systems gives rise to non-linear constraints. Recent research into non-linear decision procedures (Jha et al., 2009) makes our approach feasible.

3 Background

Our proposed algorithm builds on a number of interrelated areas including SDE models, property specifications and decision procedures. We now very briefly survey each of these topics.

3.1 Stochastic models

The behaviour of many complex systems cannot be fully captured by deterministic models such as ODEs and Boolean networks (Langmead and Jha, 2008). For these systems, non-deterministic models, such as SDEs can often provide valuable insights. Stochastic models can be broadly partitioned into discrete and continuous state categories. Examples include Discrete Time Markov Chains (DTMCs) and CTMCs. The key difference between these models lies in how they model the passage of time, as their names suggest. Continuous state space models include SDE and jump diffusion processes (Karatzas and Shreve, 1991). In such models, both the passage of time and the values of state variables are continuous. The algorithm in this paper focuses on SDE models.

An SDE (Karatzas and Shreve, 1991; Oksendal, 2003) is a differential equation in which some of the terms involve Brownian motions. A typical SDE is of the following form:

$$dX = b(t, X_t)dt + v(t, X_t)dW_t \quad (1)$$

where X is a system variable, b is a Riemann integrable function, v is an $\text{It}\bar{o}$ integrable function and W is Brownian motion.

The exact solution to an SDE is difficult to obtain. Consider the time between 0 and t as divided into m equal steps: $0, t_1, t_2, \dots, t_m = t$. Let, $\Delta = t_{k+1} - t_k$ for $0 \leq k < m$ and $W_{t_{k+1} \leftrightarrow t_k} = W_{t_{k+1}} - W_{t_k}$. Further, the solution of an SDE (Karatzas and Shreve, 1991) is the limit of the following discrete difference equation, as m goes to infinity:

$$X_{t_{k+1}} = X_{t_k} + b(t_k, X_{t_k})\Delta + v(t_k, X_{t_k})W_{t_{k+1} \leftrightarrow t_k} \quad (2)$$

It has been shown that under certain restrictions on the functions b and v (i.e. the Lipschitz condition and the linear growth condition) (Oksendal, 2003), the solution to the SDE [see equation (1)] is well defined, i.e. the limit of the above difference equations exists and is unique. Just like traditional calculus, the rules of $\text{It}\bar{o}$ integration allow us to compute closed-form solutions to *some* SDEs. As the solutions need not be deterministic, the final closed-form solution may include a Brownian motion or another process, and is itself a stochastic process. We note that our approach could be applied with greater ease on SDEs that admit a closed-form solution. In this paper, however, we will not restrict ourselves to those SDEs for which a closed-form solution can be computed, but instead present a more general approach that can be used in a larger set of cases.

3.2 Adapted finitely monitorable specifications

In this subsection, we present the formal definition of the notion of *high-level behavioural specifications* that we later use to describe the rare behaviours expected to be observed in an SDE biological model. Our interest lies in behavioural specifications

whose truth value can be decided by observing only a finite prefix of a simulation of the SDE model. The logical formulas that capture such properties are known as *Adapted Finitely Monitorable* (AFM) specifications (Jha and Langmead, 2011a; Jha and Langmead, 2011b).

A special subclass of AFM specifications on an SDE model \mathcal{M} can be expressed as formulas in *Bounded Linear Temporal Logic* (BLTL; Jha and Langmead, 2011a). For a biological SDE model \mathcal{M} , we can assume that the set of state variables V is a finite discrete-valued set of variables. A Boolean predicate over V is a constraint of the form $x \leq v$ or $x \geq v$, where $x \in V$ and $v \in \mathbb{R}$. A BLTL property is expressed on a finite set of Boolean predicates over V using Boolean and Temporal operators. The syntax and semantics of the BLTL (Pnueli, 1977; Xu and Miao, 2008) is defined by the following: $\phi ::= x \leq v | x \geq v | (\phi_1 \vee \phi_2) | (\phi_1 \wedge \phi_2) | \neg \phi_1 | (\phi_1 U^t \phi_2) | X \phi_1$.

Here $x \in V$, $v \in \mathbb{Q}$ and $t \in \mathbb{R}^+$. Additionally, we can define other temporal operators such as: $G^t \psi = \neg F^t \neg \psi$ or $F^t \psi = \text{True} U^t \psi$ in terms of the bounded until operators (U^t). Explicitly, the formula $G^t \psi$ implies that ψ holds at all moments for the next t time units into the future and the formula $F^t \psi$ implies that ψ holds sometime within the next t time units.

The fact that a path σ of a stochastic model satisfies property ϕ is expressed by $\sigma \models \phi$. Assume $\sigma = (s_0, \Delta_0), (s_1, \Delta_1), \dots$ be a trace of the model along s_0, s_1, \dots states with durations $\Delta_0, \Delta_1, \dots \in \mathbb{R}^+$. We define the path that starts at state i by σ^i (explicitly, σ^0 refers the original execution σ). $V(\sigma, i, x)$ denotes the value of the state variable x in σ at the state i . The semantics of BLTL with respect to the paths of \mathcal{M} explained below:

- $\sigma^k \models x \leq v$ iff $V(\sigma, k, x) \leq v$: i.e. the constraint $x \leq v$ is true in state s_k ;
- $\sigma^k \models x \geq v$ iff $V(\sigma, k, x) \geq v$: i.e. the constraint $x \geq v$ is true in state s_k ;
- $\sigma^k \models \phi_1 \wedge \phi_2$ iff $\sigma^k \models \phi_1$ and $\sigma^k \models \phi_2$: i.e. ϕ_1 and ϕ_2 are both true in state s_k ;
- $\sigma^k \models \phi_1 \vee \phi_2$ iff $\sigma^k \models \phi_1$ or $\sigma^k \models \phi_2$: i.e. ϕ_1 or ϕ_2 is true in state s_k ;
- $\sigma^k \models \neg \phi_1$ iff $\sigma^k \models \phi_1$ does not hold: i.e. ϕ_1 is not true in state s_k ;
- $\sigma^k \models \phi_1 U^t \phi_2$ iff there exists $i \in \mathbb{N}$ such that (a) $0 \leq \sum_{j=k}^{k+i} \Delta_j \leq t$, (b) $\sigma^{k+i} \models \phi_2$, and (c) for each $0 \leq j < i$, $\sigma^{k+j} \models \phi_1$: i.e. within t time, ϕ_2 will be true and ϕ_1 will hold until then.
- $\sigma^k \models X \phi_1$ iff $\sigma^{k+1} \models \phi_1$: i.e. ϕ_1 has to be true in state s_{k+1} .

3.3 Bit-vector decision procedure

Decision procedures are algorithms that can be used to decide whether a given logical formula (such as those discussed in the previous section) can have a satisfying assignment under a given interpretation of its constituent variables (Kroening and Strichman, 2008). Unlike incomplete theorem provers for higher order logic formulas

(Nipkow et al., 2002), algorithms for deciding the existential fragment of bit-vector first order logic (Jackson, 2000) are efficient and practical (Ghosh, 2010; Jha et al., 2009). Traditionally, two approaches have been used to solve bit-vector arithmetic constraints. In one approach, bit-vectors are translated into Boolean propositions by a process called ‘bit blasting’ (Ganesh and Dill, 2007). This approach is similar to the process by which VLSI circuits for arithmetic are built. In another approach, the decision procedure uses a mathematical programming engine to reason about conjunctions of constraints in the bit-vector SMT (Biere et al., 2009) formula. Here, the decision procedure may make multiple calls to a mathematical programming engine which, when combined with efficient bookkeeping via the Davis–Putnam–Logemann–Loveland (DPLL) algorithm, can be used to decide the truth value of the formula. SMT solvers can determine the satisfiability of formulas in expressive logics such as the existential fragment of first-order logic and reason about background theories such as non-linear arithmetic that fix the interpretations of certain predicates and function’s symbols. Our algorithm uses an SMT-based approach to reason about SDEs.

Several existing tools including Beaver (Jha et al., 2009), Boolector (Brummayer and Biere, 2009), CVC3 (Barrett and Tinelli, 2007), Yices (Dutertre and de Moura, 2006) and Z3 (de Moura and Bjørner, 2008) can be used to reason about bit-vector SMT formulas. SMT solvers like Beaver have been used to study non-linear hybrid systems and complex software systems with non-linear constraints (Gulwani et al., 2011; Jha et al., 2010a; Jha et al., 2010b; Jha et al., 2007). Bit-vector decision procedures like Beaver can also analyse the feasibility of *non-linear constraints with fixed-precision* arithmetic connected by arbitrary Boolean connectives like AND, OR and NOT. The ability to analyse non-linear arithmetic constraints connected by Boolean connectives other than conjunction is a key feature of non-linear bit-vector decision procedures.

In the context of our algorithm, the bit-vector decision procedure is invoked to search for suitable values of the Brownian motion at various points of time in a simulation such that (a) the SDE model shows the behaviour in the given specification and (b) the overall probability density of observing the discrete behaviour is maximised. Our algorithm may invoke the decision procedure several times to ensure that these high level constraints are met.

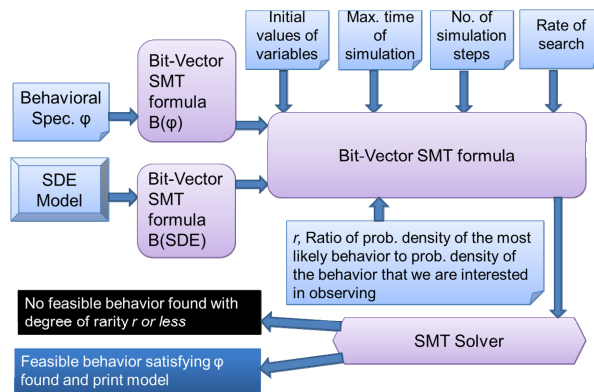
4 Algorithms and results

In this section, we present our algorithm and key theoretical results that prove the correctness of the algorithm. The high-level idea behind the algorithm is that it transforms both the SDE model and the behavioural specification into bit-vector SMT formulas, and then uses a decision procedure to decide whether the conjunction of all the formulas can be satisfied (up to a fixed precision). If the formulas cannot be satisfied, then the model does not exhibit the behaviour (again, up to fixed precision). If the formulas can be satisfied, the algorithm returns a discrete witness trajectory with maximal likelihood. Figure 2 provides a sketch of our proposed algorithm.

Our correctness proof relies on the independence property of increments in Brownian motion stochastic processes and in the Gaussian distribution of the increments themselves. Intuitively, high probability density behaviours must correspond to small

Brownian motion increments. However, in this section, we show that this intuition is only partly true – high probability density behaviours correspond to small *sums of squares of Brownian motion increments*.

Figure 2 A sketch of our decision procedure-based algorithm that investigates the likelihood of rare behaviours in stochastic differential equation models. Our algorithm translates the stochastic differential equation model and the behavioural specification into bit-vector SMT formulas. It uses an SMT-solver to explore rare behaviours given as input in the specification (see online version for colours)



4.1 Algorithm

Our algorithm is shown in Figure 3. The algorithm has seven inputs:

- 1 Stochastic differential equation: $dX(t)=b(t, X_t)dt+v(t, X_t)dW_t$
- 2 Initial state for the solution to the SDE: $X(0)$
- 3 Finitely monitorable behavioural specification: ϕ
- 4 Maximum time for model simulation: t_m
- 5 Rate of search: $\alpha, 0 < \alpha < 1$
- 6 Number of discrete time steps in the numerical simulation: m
- 7 Ratio of probability density of the most likely behaviour to the probability density of the behaviour that we are interested in observing: r .

The algorithm produces a discrete solution of the SDE that also satisfies the given behavioural specification. The algorithm uses the maximum time for model simulation t_m and the number of discrete time steps m to compute the temporal discretisation Δ at which the SDE solution should be sampled and computes the threshold T_{max} on the sum of the squares of the Brownian increments using a result that we derive in Lemma 5. Then, the SDE model and the behavioural specification are transformed into SMT bit-vector constraints. Finally, the SMT solver is used to obtain a possible solution \bar{X} to the constraints enforced by the SDE, the initial condition, the bound on the probability, and the rare behaviour we want to observe. If such behaviour does not exist, the algorithm

stops and reports the absence of such behaviour. If the algorithm finds a witness, it continues searching for behaviours with higher probability by reducing the threshold that bounds the sum of the squares of the Brownian motion increments.

Figure 3 Algorithm for discovering rare behaviours in stochastic models. Our algorithm uses the bit-vector decision procedures for verifying stochastic models with respect to a given behavioural specification. It either produces a discrete solution path of the stochastic differential equation satisfying the specification or reports the absence of the specified behaviour (up to a user-specified confidence)

Require: $dX(t) = b(t, X_t)dt + v(t, X_t)dW_t$: Stochastic differential equation
 $X(0)$: Initial value of the variable X
 ϕ : Behavioral specification in a finitely monitorable logic like BLTL
 t_m : Maximum time of simulation
 α : Rate of search; $0 < \alpha < 1$ (for a fine search, choose an α close to 1)
 m : Number of discrete simulation steps
 r : Ratio of probability density of the most likely behavior to the probability density of behavior we want to observe

Ensure: Produce a behavior τ satisfying ϕ with probability density at least P_{min} (a function of m and r) or reports the absence of the desired behavior

$\Delta = \frac{t_m}{m}$
 P_{min} : The minimum value of the desired probability density of the behavior under investigation, when r is given.
 $P_{min} = \frac{(2\pi)^{-m/2} \Delta^{-m/2}}{r}$ (using Lemma 2)
Threshold $T_{max} = 2\Delta \log\left(\frac{1}{P_{min}(2\pi\Delta)^{m/2}}\right)$
 $\mathcal{B}(\phi) \leftarrow \text{Specification2BooleanAssertion}(\phi)$ //Transform specification ϕ to Boolean assertions.
//Transform SDE to Boolean assertions
 $\mathcal{B}(SDE) \leftarrow \bigwedge_{k=0}^{m-1} (X_{t_{k+1}} = X_{t_k} + b(t_k, X_{t_k}) \Delta + v(t_k, X_{t_k}) W_{t_{k+1} \leftrightarrow t_k})$
 $T = T_{max}$ //Initialize threshold
 $\mathcal{B}(Init) \leftarrow (X_0 == X(0))$ //Assert initial value
FeasibleWitness = 0
while $T \geq 0$ **do**
 $\mathcal{B}(Threshold) \leftarrow \left(\sum_{k=0}^{m-1} (W_{t_{k+1} \leftrightarrow t_k})^2 \leq T\right)$ //Assert probability density of rare behavior
//The following expression is evaluated by calling SMT solver
if $\mathcal{B}(Init) \wedge \mathcal{B}(SDE) \wedge \mathcal{B}(\phi) \wedge \mathcal{B}(Threshold)$ **then**
//A feasible solution has been obtained
 $[\bar{X}, \bar{W}_{t_{k+1} \leftrightarrow t_k}] \leftarrow \text{Feasible model for } \mathcal{B}(Init) \wedge \mathcal{B}(SDE) \wedge \mathcal{B}(\phi) \wedge \mathcal{B}(Threshold)$
FeasibleWitness = 1
 $P = (2\pi)^{-m/2} \Delta^{-m/2} e^{-\frac{\sum_{k=0}^{m-1} (W_{t_{k+1} \leftrightarrow t_k})^2}{2\Delta}}$ //Compute probability density
else
break
end if
 $T = \alpha T$
end while
if FeasibleWitness == 0 **then**
Print "No feasible behavior found with degree of rarity r or less"
else
Print "Feasible behavior satisfying ϕ found"
Print model behavior $[\bar{X}, \bar{W}_{t_{k+1} \leftrightarrow t_k}]$
Print $\frac{(2\pi)^{-m/2} \Delta^{-m/2}}{P}$ //This ratio indicates the observed behavior is how many times less likely than the most likely behavior.
end if

4.1.1 SDE to bit-vector SMT formula

Recall the solution [equation (2)] to an SDE is the limit of the following discrete equation (as m goes to infinity):

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

In our approach, we represent each of the variables X_{t_k} and $W_{t_{k+1} \leftrightarrow t_k}$ using fixed-precision bit-vectors and use bit-vector arithmetic to reason about mathematical operations. The solution to the SDE can be represented by the conjunction of the above constraints as: $\bigwedge_{k=0}^{m-1} \left(X_{t_{k+1}} = X_{t_k} + b(t_k, X_{t_k}) \Delta + v(t_k, X_{t_k}) W_{t_{k+1} \leftrightarrow t_k} \right)$. In order to ensure that the fixed precision does not lead to stability issues, we consider bit-vectors with varying orders of bit-width. Also, we add constraints to include the possible set of initial values of the variable X .

Our use of fixed-precision bit-vector arithmetic to reason about SDEs is similar to the study of dynamical hybrid systems with non-linear arithmetic constraints (Jha et al., 2007). However, use of the decision procedure is limited to the exhaustive analysis of non-deterministic choices made by a possibly adversarial agent while we study the probabilistic outcome of a stochastic process. By adding further constraints to the SMT formula representing the solutions to the SDE, we restrict the solution space of the SDE to those behaviours that actually satisfy the given specification ϕ . Also, by carefully adding a well-thought constraint on the magnitudes of the bit-vector variables representing the Brownian motion, we enforce the constraint that these discrete behaviours have a high probability density by minimising the *sum of squares of Brownian motion increments*.

4.1.2 Specification to bit-vector SMT formula

In order to constrain the behaviours of the SDE, we translate the finitely monitorable specification to an SMT formula over bit-vectors. In this subsection, we demonstrate the translation of BLTL formulas into a bit-vector formula. The translation function \mathcal{B} takes a BLTL formula and a time step as input. Our translation is given by the following recursive rewrite rules:

- $\mathcal{B}(X \geq v, k) = X_k \geq v$;
- $\mathcal{B}(X \leq v, k) = X_k \leq v$;
- $\mathcal{B}(\phi_1 \wedge \phi_2, k) = \mathcal{B}(\phi_1, k) \wedge \mathcal{B}(\phi_2, k)$;
- $\mathcal{B}(\neg \phi, k) = \neg \mathcal{B}(\phi, k)$;
- $\mathcal{B}(\phi_1 U^\tau \phi_2, k) = \bigvee_{i=k}^{m-k} \left(0 \leq \sum_{j=k}^{k+i} \Delta \leq \tau \wedge \mathcal{B}(\phi_2, k+i) \wedge \left(\bigwedge_{j=0}^{i-1} \mathcal{B}(\phi_1, k+j) \right) \right)$.

We note that the right-hand side of every rewrite rule is either a bit-vector SMT formula or a Boolean combination of a bit-vector SMT formula and the translation function \mathcal{B} over a simpler formula. Because our specification formulas have finite sizes and we are only interested in properties over bounded time, our translation is bound to terminate and produce a finite bit-vector SMT formula. One can prove this using induction on the length of the formula and the time step parameter in the definition of the translation function.

4.2 Proof of correctness

In this section, we argue the correctness of our algorithm (see Figure 3). When the algorithm produces a likely discrete behaviour of the SDE satisfying the given

specification, ϕ , we derive the probability density of the discrete behaviour obtained. Our results rely on the independence of increments of Brownian motion and their Gaussian distribution. Recall the Brownian motion W (Oksendal, 2003) is a continuous-time stochastic process satisfying the following three conditions:

- 1 $W_0 = 0$
- 2 W_t is continuous (*almost surely*).
- 3 W_t has independent *normally distributed* increments:
 - $W_t - W_s$ and $W_{t'} - W_{s'}$ are independent if $0 \leq s < t < s' < t'$.
 - $W_t - W_s$ is distributed as $\mathcal{N}(0, t-s)$, where $\mathcal{N}(0, t-s)$ denotes the normal distribution with mean 0 and variance $t-s$.

4.2.1 Probability density of solution to SDE

Given a finely discretised solution $X_{t_0}, X_{t_1}, \dots, X_{t_m}$ to the SDE initial value problem, we want to determine the probability of observing this solution. Given the initial value X_{t_0} , we want to compute the probability density of observing the value X_{t_1} after t_1 time. Suppose the corresponding values of Brownian motion at t_0 and t_1 are, respectively, \widehat{W}_{t_0} and \widehat{W}_{t_1} . Then:

$$P(X_{t_1} | X_{t_0}) = P(W_{t_1} - W_{t_0} = \widehat{W}_{t_1} - \widehat{W}_{t_0} | X_{t_0}) \tag{3}$$

$$= P(W_{t_1} - W_{t_0} = \widehat{W}_{t_1} - \widehat{W}_{t_0} | W_{t_0} = \widehat{W}_{t_0}) \tag{4}$$

$$= \frac{1}{\sqrt{2\pi(t_1 - t_0)}} e^{-\left(\frac{\widehat{W}_{t_1} - \widehat{W}_{t_0}}{2(t_1 - t_0)}\right)^2} \tag{5}$$

Equation (3) holds because the only stochastic component of X_{t_1} is completely determined given the increment in Brownian motion and the initial value X_{t_0} for small values of $t_1 - t_0$. Further, X_t is adapted to the stochastic process W_t , and hence [equation (4)] is true. Since, increments in Brownian motion are normally distributed, as we know $\widehat{W}_{t_1} - \widehat{W}_{t_0} \sim \mathcal{N}(0, t_1 - t_0)$. Hence, equation (5) gives the desired probability distribution.

Lemma 1: *The probability density of a discretised solution to a stochastic differential equation is inversely proportional to the exponential of the sum of squares of the increments of the Brownian motion.*

Proof 1: We compute the probability density of observing the sequence of the observed discretised solution given the initial value:

$$\begin{aligned}
 & P(X_{t_0}, X_{t_1}, \dots, X_{t_m} | X_{t_0}) \\
 &= P(X_{t_1}, \dots, X_{t_m} | X_{t_0}) P(X_{t_0} | X_{t_0}) \\
 &= P(X_{t_1}, X_{t_2}, \dots, X_{t_m} | X_{t_0}) \quad \text{Since, } P(X_{t_0} | X_{t_0})=1 \\
 &= P(X_{t_2}, \dots, X_{t_m} | X_{t_0}, X_{t_1}) P(X_{t_1} | X_{t_0}) \\
 &= P(X_{t_2}, \dots, X_{t_m} | X_{t_1}) P(X_{t_1} | X_{t_0}) \tag{6} \\
 &\text{Since, } P(X_{t_2} | X_{t_0}, X_{t_1}) = P(X_{t_2} | X_{t_1}) \\
 &= P(X_{t_3}, \dots, X_{t_m} | X_{t_1}, X_{t_2}) P(X_{t_2} | X_{t_1}) P(X_{t_1} | X_{t_0}) \\
 &= \dots \dots \dots \\
 &= P(X_{t_m} | X_{t_{m-1}}) \dots P(X_{t_1} | X_{t_0}) = (2\pi)^{-m/2} \Delta^{-m/2} e^{-\frac{1}{2\Delta} \left(\sum_{i=1}^m (\widehat{W}_i - \widehat{W}_{t_{i-1}})^2 \right)}
 \end{aligned}$$

Conditional independence has been used repeatedly in the above derivation.

Equation (6) gives the probability density of observing a given behaviour in an SDE. It satisfies our intuition that large values of Brownian motion increments should correspond to smaller probability densities while small values of Brownian motion increments correspond to large probability densities. However, equation (6) shows that the probability density only depends on the sum of squares of the increments of Brownian motion, and not on the individual increments themselves.

Lemma 2: The highest probability density possible for a behaviour is $(2\pi)^{-m/2} \Delta^{-m/2}$.

Proof 2:

$$P = (2\pi)^{-m/2} \Delta^{-m/2} e^{-\frac{1}{2\Delta} \left(\sum_{i=1}^m (\widehat{W}_i - \widehat{W}_{t_{i-1}})^2 \right)}$$

P will be highest when $\sum_{i=1}^m (\widehat{W}_i - \widehat{W}_{t_{i-1}})^2 = 0$ and the highest value is $(2\pi)^{-m/2} \Delta^{-m/2}$.

Lemma 3: $\sum_{i=1}^m \left(\frac{\widehat{W}_i - \widehat{W}_{t_{i-1}}}{\sqrt{\Delta}} \right)^2$ has a χ -distribution with m degrees of freedom.

Proof 3: As $\frac{\widehat{W}_i - \widehat{W}_{t_{i-1}}}{\sqrt{\Delta}}$ are independent normal random variables with unit variance and

mean 0, $\sum_{i=1}^m \left(\frac{\widehat{W}_i - \widehat{W}_{t_{i-1}}}{\sqrt{\Delta}} \right)^2$ has a χ -distribution with m degrees of freedom.

Lemma 4: Given the sum of squares of the magnitudes of the Brownian motion increments, the probability of a path is independent of the Brownian motions themselves.

Proof 4: The above result follows from Lemma 3, and rewriting equation (6).

Lemma 5: Given the probability density P of a discretised SDE behaviour with m samples every Δ time apart, the sum of squares of increments of Brownian motion

$$T \equiv \sum_{i=1}^m (\widehat{W}_{t_i} - \widehat{W}_{t_{i-1}})^2 \text{ should be equal to } 2\Delta \log \left(\frac{1}{P(2\pi\Delta)^{m/2}} \right).$$

Proof 5:

$$\begin{aligned} P &= (2\pi)^{-m/2} \Delta^{-m/2} e^{-\frac{1}{2\Delta} \left(\sum_{i=1}^m (\widehat{W}_{t_i} - \widehat{W}_{t_{i-1}})^2 \right)} \quad \text{From equation (6)} \\ &\Rightarrow P = (2\pi)^{-m/2} \Delta^{-m/2} e^{-\frac{1}{2\Delta} T} \quad \text{Definition of } T \\ &\Rightarrow \frac{P}{(2\pi)^{-m/2} \Delta^{-m/2}} = e^{-\frac{1}{2\Delta} T} \\ &\Rightarrow \frac{P(2\pi)^{m/2}}{\Delta^{-m/2}} = e^{-\frac{1}{2\Delta} T} \quad (7) \\ &\Rightarrow \log \left(\frac{P(2\pi)^{m/2}}{\Delta^{-m/2}} \right) = -\frac{1}{2\Delta} T \\ &\Rightarrow \log \left(\frac{\Delta^{-m/2}}{P(2\pi)^{m/2}} \right) = \frac{1}{2\Delta} T \\ &\Rightarrow T = 2\Delta \log \left(\frac{1}{P(2\pi\Delta)^{m/2}} \right) \end{aligned}$$

A key concern in discretising a continuous SDE is the error introduced by sampling a continuous system, and replacing an SDE with a discretised difference equation. The existence and uniqueness of SDE ensures that the solution of a sufficiently discretised SDE approaches the solution of the continuous SDE (Kloeden and Platen, 2011).

5 Experimental results

In this section, we explore an SDE model representing the dynamics of the eukaryotic cell cycle (CC). We used the cell cycle model developed by Pfeuty and Kaneko (2007). The model describes the dynamics of cell growth and cell cycle division. We rewrite equations from their model (Pfeuty and Kaneko, 2007) to emphasise their stochastic nature:

$$\begin{aligned} dX(t) &= (s_x - (d_x + c_{xy} Y) X + a_x X^2) dt + \sigma_x d\zeta_x(t) \\ dY(t) &= (-d_y Y + s_y X^2) dt + \sigma_y d\zeta_y(t) \\ dV(t) &= gV dt + \sigma_v d\zeta_v(t) \end{aligned}$$

where X represents a set of components that are involved in triggering cell-cycle events (Pfeuty and Kaneko, 2007). Y represents a set of components that impose restrictions on

the activity of X -components (Pfeuty and Kaneko, 2007). V is the size of the cell and ζ_x , ζ_y and ζ_v represent standard Brownian motions (Pfeuty and Kaneko, 2007). A cell divides when the concentration of the Y -component reaches a critical value ($\theta = 2$) (Pfeuty and Kaneko, 2007). For our experiments, we set the normal time between cell divisions as 33 time units, and the normal cell size at cell division time as 2.88 units.

We apply our decision procedure-based algorithm to the model and investigate the likelihood of unlikely behaviours (including the rare behaviours) of the model. Such behaviours in the cell cycle can cause cytokinetic diseases including cancers and tumours. We recall that due to the probabilistic flavour of verification employed by our technique for discovering rare behaviours in stochastic models, our algorithm takes as input a number that represents the *degree of rarity* (i.e. the ratio of the probability density of the most likely behaviour of the model to the probability density of the behaviour that we are interested in investigating).

We use our algorithm to determine if it is possible to decrease the time between cell divisions by at least 15% of the normal cell division time. The following BLTL specification captures this behavioural specification: $F^{28.05}(Y \geq 2)$. Our algorithm finds that the above behaviour is possible, and it is 10^{20} times less likely than the most likely behaviour. It is difficult to explore such unlikely behaviour using simulations. If one simulation run takes 1 millisecond, then it may take 10^{17} seconds, i.e. $3.17 \cdot 10^9$ years to produce this behaviour.

We also investigate if it is possible to decrease the time between cell divisions by at least 10% of the normal cell division time (in BLTL, $F^{29.7}(Y \geq 2)$), and 5% of the normal cell division time (in BLTL, $F^{31.35}(Y \geq 2)$). We find that decreasing time between cell divisions by 10% is 10^6 times less likely than the most likely behaviour. We verify using simulations and produce the sample paths for Y which satisfy the behaviour (see Figure 4). Figure 5 shows sample paths for Y when the time between cell divisions is decreased by 5% of the normal cell division time.

Figure 4 Simulation results for decreasing the time between cell divisions by at least 10% of the normal cell division time. We know that the cell divides when Y reaches the threshold value of 2. In the figure, the value of Y is plotted against time. Sample paths that satisfy the behaviour are shown in the plot. The observed behaviour is 10^6 times less likely than the most likely behaviour (see online version for colours)

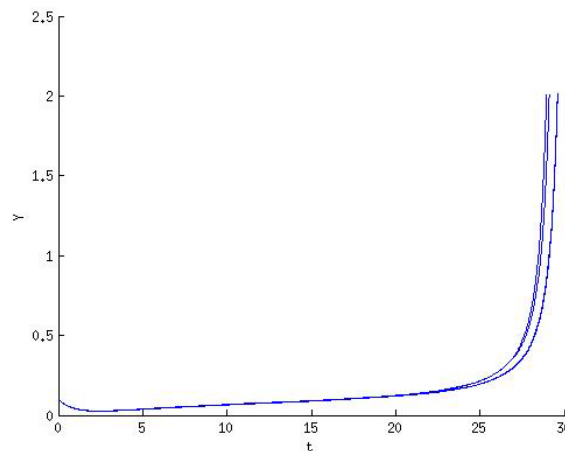
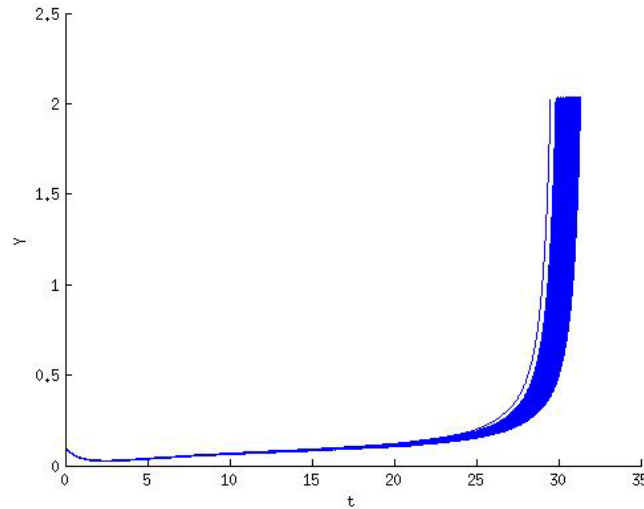


Figure 5 Simulation results for decreasing the time between cell divisions by at least 5% of the normal cell division time. The observed behaviour is 18 times less likely than the most likely behaviour. Sample paths that satisfy the behaviour are shown in the figure (see online version for colours)



We also use our algorithm to explore the likelihood of unlikely behaviours in cell size. We consider the following behaviour: is it possible to increase cell size by at least 30% of the normal cell size within the normal cell division time. This property can be encoded as following BLTL specification: $F^{33}(V \geq 3.74)$. Our algorithm reports that the model is incapable of producing the stated behaviour with degree of rarity 10^{11} .

We also investigate the following: is it possible to increase cell size by at least 10% of the normal cell size within the normal cell division time (in BLTL, $F^{33}(V \geq 3.17)$), and 5% of the normal cell size within the normal cell division time (in BLTL, $F^{33}(V \geq 3.02)$). Our algorithm finds that increasing cell size by 10% of the normal cell size is 10^{11} times less likely than the most likely behaviour, but increasing cell size by 5% is 854 times less likely than the most likely behaviours. Figure 6 shows sample paths for V when the cell size is increased by 5% of the normal cell size. The results of our experiments are shown in Table 1.

Given a stochastic model and a temporal logic behavioural specification, our algorithm attempts to find the existence of rare behaviours in the model – it either reports the absence of the desired behaviour up to the initial fixed degree of rarity, or produces a witness trajectory of the behaviour being considered along with the maximum possible likelihood of that behaviour. Using pure numerical simulations, it is almost impossible to observe such rare behaviours or even prove their absence. Sampling is an inefficient means for studying rare behaviours because the vast majority of sampled trajectories will not exhibit desired behaviours. Our algorithm uses SMT-based symbolic decision procedures to explore all possible behaviours of the stochastic biological models without explicitly enumerating them.

Figure 6 Simulation results for increasing cell size by at least 5% of the normal cell size within the normal cell division time. In the figure, we show how the cell size (V) changes with time. The observed behaviour is 854 times less likely than the most likely behaviour. Sample paths that satisfy the behaviour are shown in the figure

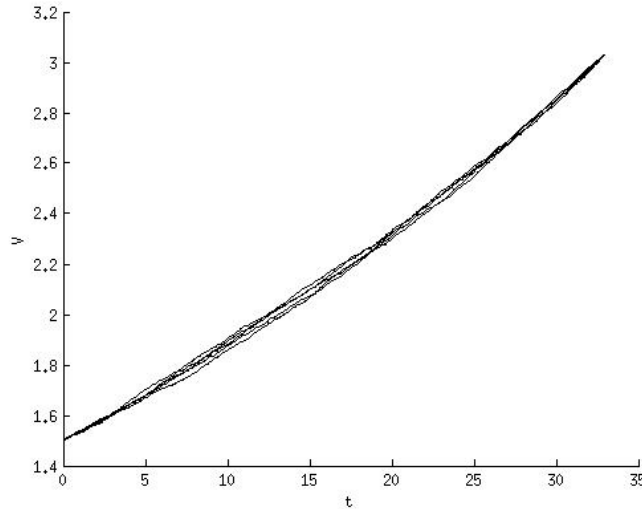


Table 1 Verification of feasibility of various cell division times and cell size changes. This table describes the results of the experiments about the time between cell divisions and cell size change in the CC model described in the experiment’s section

Decrease time between cell divisions			Increase cell size		
Time change (%)	Ratio of probability density of most likely behaviour to probability density of observed behaviour	Simulation verification	Size change (%)	Ratio of probability density of most likely behaviour to probability density of observed behaviour	Simulation verification
5%	18	Yes	5%	854	Yes
10%	10^6	Yes	10%	10^{11}	No
15%	10^{20}	No	30%	10^{11} (Not feasible)	No

6 Conclusion

We have introduced an algorithm for efficiently investigating rare behaviours in SDE models. Informally, our method avoids the computational costs associated with sampling by, in effect, searching for trajectories from the model that satisfies a given behavioural specification. That is, our method only generates trajectories that exhibit the behaviour (if such trajectories exist), and then estimates the probability density of those trajectories. The actual search is performed by converting the SDE and the behavioural specification into bit-vector SMT formulas, and then calling an appropriate decision procedure to find witness trajectories. Thus, our method takes advantage of the efficiency and power of modern SMT-solvers. Consequently, as newer, more powerful decision procedures are created, our method will inherit the benefits of those methods.

While it can be argued that use of fixed-precision decision procedures is a limiting factor, it can be shown that the errors introduced through discretisation can be bounded. More importantly, our method enables users to explicitly specify the desired level of accuracy (in terms of the number of bits). Increased precision can be obtained by increasing the number of bits. Moreover, we feel that fixed-precision answers are usually sufficient in many soft applications (such as weather prediction model and epidemiological model) since the computational models themselves are merely abstractions of reality, and will be compared to experimental data which also has fixed precision.

Several interesting directions for future research remain open. We are studying the use of SMT solving techniques to analyse rare behaviours of closed-form solutions to SDEs. Many practical applications require the study of a system where one component is an SDE and the other component may be an ODE or even a finite state controller. Our proposed rare behaviour discovery approach should be extended to such systems in order to study biologically important cyber-physical systems like artificial pancreas. The development of specialised SMT solving techniques that can solve the decision problems arising from such analysis is also an exciting area.

Acknowledgements

The authors thank Emily Sassano for comments on a draft of this paper. The authors also thank the anonymous reviewers of the 2nd IEEE International Conference on Computational Advances in Bio and Medical Sciences (ICCABS 2012) for their comments and suggestions.

References

- Arato, M. (2003) 'A famous nonlinear stochastic equation (Lotka-Volterra model with diffusion)', *Mathematical and Computer Modelling*, Vol. 38, No. 7, pp.709–726.
- Baier, C. and Katoen, J.P. (2008) *Principles of Model Checking (Representation and Mind Series)*, MIT Press.
- Barrett, C. and Tinelli, C. (2007) 'CVC3', in Damm, W. and Hermanns, H. (Eds): *Proceedings of the 19th International Conference on Computer Aided Verification (CAV'07)*, Springer, Berlin, Germany, pp.298–302.
- Biere, A., Heule, M.J.H, van Maaren, H. and Walsh, T. (Eds) (2009) *Handbook of Satisfiability*, Vol. 185, IOS Press.
- Brummayer, R. and Biere, A. (2009) 'Boolector: an efficient SMT solver for bit-vectors and arrays', *Proceedings of the 15th International Conference on Tools and Algorithms for the Construction and Analysis of Systems: Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2009, in TACAS '09*, Springer-Verlag, Berlin, Heidelberg, pp.174–177.
- Clarke, E.M., Faeder, J.R., Langmead, C.J., Harris, L.A., Jha, S.K. and Legay, A. (2008) 'Statistical model checking in biolab: applications to the automated analysis of T-cell receptor signaling pathway', *Proceedings of the 6th International Conference on Computational Methods in Systems Biology in CMSB '08*, Springer-Verlag, Berlin, Heidelberg, pp.231–250.

- Cohen, E., Dahlweid, M., Hillebrand, M., Leinenbach, D., Moskal, M., Santen, T., Schulte, W. and Tobies, S. (2009) 'VCC: a practical system for verifying concurrent C', *Theorem Proving in Higher Order Logics*, pp.23–42.
- de Jong, H. (2002) 'Modeling and simulation of genetic regulatory systems: a literature review', *Journal of Computational Biology*, Vol. 9, pp.67–103.
- de Moura, L. and Bjørner, N. (2008) 'Z3: an efficient SMT solver', *Tools and Algorithms for the Construction and Analysis of Systems*, pp.337–340.
- Dutertre, B. and de Moura, L. (2006) *The Yices SMT Solver*, Tool Paper. Available online at: <http://yices.csl.sri.com/tool-paper.pdf>
- Ganesh, V. and Dill, D. (2007) 'A decision procedure for bit-vectors and arrays', *Computer Aided Verification*, Springer, pp.519–531.
- Ghosh, A.K. (2010) 'Speeding up SAT solver by exploring CNF symmetries: revisited', *Congressus Numerantium*, Vol. 206, pp.73–84.
- Ghosh, A.K., Hussain, F., Jha, S.K., Langmead, C.J. and Jha, S. (2012) 'Decision procedure based discovery of rare behaviors in Stochastic differential equation models of biological systems', *IEEE 2nd International Conference on Computational Advances in Bio and Medical Sciences (ICCABS)*, pp.1–6.
- Gulwani, S., Jha, S., Tiwari, A. and Venkatesan, R. (2011) 'Synthesis of loop-free programs', *ACM SIGPLAN Notices*, Vol. 46, pp.62–73.
- Heath, J., Kwiatkowska, M., Norman, G., Parker, D. and Tymchyshyn, O. (2006) 'Probabilistic model checking of complex biological pathways', in Priami, C. (Ed.): *Proceedings of Computational Methods in Systems Biology (CMSB'06)*, Springer Verlag, pp.32–47.
- Jackson, D. (2000) 'Automating first-order relational logic', *ACM SIGSOFT Software Engineering Notes*, ACM, pp.130–139.
- Jha, S.K., Brady, B.A. and Seshia, S.A. (2007) 'Symbolic reachability analysis of lazy linear hybrid automata', *FORMATS*, pp.241–256.
- Jha, S.K., Clarke, E.M., Langmead, C.J., Platzer, A. and Zuliani, P. (2009) 'A Bayesian approach to model checking biological systems', *Proceedings of 7th Conference on Computational Methods in Systems Biology (CMSB)*, pp.218–234.
- Jha, S.K., Gulwani, S., Seshia, S. and Tiwari, A. (2010a) 'Oracle-guided component-based program synthesis', *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering*, ACM, New York, NY, USA, pp.215–224.
- Jha, S.K. and Langmead, C.J. (2011a) 'Exploring behaviors of SDE models of biological systems using change of measures', *BMC Bioinformatics*, Vol. 13, doi:10.1186/1471-2105-13-S5-S8.
- Jha, S.K. and Langmead, C.J. (2011b) 'Synthesis and infeasibility analysis for stochastic models of biochemical systems using statistical model checking and abstraction refinement', *Theoretical Computer Science*, Vol. 412, No. 21, pp.2162–2187.
- Jha, S.K., Limaye, R. and Seshia, S.A. (2009) 'Beaver: engineering an efficient SMT solver for bit-vector arithmetic', *CAV*, pp.668–674.
- Jha, S.K., Seshia, S.A. and Tiwari, A. (2010b) 'Synthesizing switching logic for safety and dwell-time requirements', *Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems in ICCPS'10*, ACM, New York, NY, USA, pp.22–31.
- Juneja, S. and Shahabuddin, P. (2006) 'Rare event simulation techniques: an introduction and recent advances', in Henderson, S. and Nelson, B. (Eds): *Handbook on Simulation*, pp.291–350.
- Karatzas, I. and Shreve, S. (1991) *Brownian Motion and Stochastic Calculus (Graduate Texts in Mathematics)*, 2nd ed., Springer.
- Kloeden, P.E. and Platen, E. (2011) *Numerical Solution of Stochastic Differential Equations*, Springer.
- Kroening, D. and Strichman, O. (2008) *Decision Procedures – An Algorithmic Point of View of EATCS*, Springer.

- Langmead, C.J. and Jha, S.K. (2008) 'Symbolic approaches to finding control strategies in Boolean networks', *Proceedings of 6th Asia Pacific Bioinformatics Conference (APBC)*, pp.307–320.
- Lefever, R. and Garay, R. (1978) *Local Description of Immune Tumor Rejection*, Elsevier/North-Holland Biomedical Press, pp.333–344.
- Marwedel, P. (2008) *Embedded System Design: Embedded Systems Foundations of Cyber-Physical Systems*, Springer.
- Nipkow, T., Paulson, L.C. and Wenzel, M. (2002) *Isabelle/HOL: A Proof Assistant for Higher-Order Logic*, Springer, Vol. 2283.
- Oksendal, B. (2003) *Stochastic Differential Equations: An Introduction with Applications*, 6th ed., Springer.
- Pfeuty, B. and Kaneko, K. (2007) 'Minimal requirements for robust cell size control in eukaryotic cells', *Physical Biology*, Vol. 4, No. 3, pp.194–204.
- Piazza, C., Antoniotti, M., Mysore, V., Policriti, A., Winkler, F. and Mishra, B. (2005) 'Algorithmic algebraic model checking I: challenges from systems biology', *CAV*, pp.5–19.
- Pnueli, A. (1977) 'The temporal logic of programs', *Annual IEEE Symposium on Foundations of Computer Science*, pp.46–57.
- Xu, Q.G. and Miao, H.K. (2008) 'Timed automata for metric interval temporal logic formulae in prototype verification system', *Journal of Shanghai University (English Edition)*, Vol. 12, No. 4, pp.339–346.