

A Probabilistic Calculus for Molecular Systems

Roberto Barbuti, Stefano Cataudella, Andrea Maggiolo-Schettini,
Paolo Milazzo, and Angelo Troina

Dipartimento di Informatica, Università di Pisa
Via Buonarroti 2, 56127 - Pisa, Italy
{*barbuti, cataudel, milazzo, maggiolo, troina*}@di.unipi.it

Abstract. We introduce a calculus for molecular reactions based on probabilistic rules. On the one hand, according to the semantics of the rules, we give an interpreter to study the behaviour of molecular systems. On the other hand, we model molecular reactions through a probabilistic transition system. This allows us to check properties of molecular systems by using, for example, the PRISM model-checker.

1 Introduction

In the past few years people have become aware that biological processes can be described using means originally developed by computer scientists to model systems of interacting components. This permits simulation of system behaviour and verification of properties. Hence, many formalisms have been developed to describe biological phenomena at different levels. In particular, several formalisms have been proposed to describe biomolecular and membrane interactions [4, 7, 6, 8, 9, 12, 13, 16, 17, 19].

In this paper we introduce a probabilistic calculus for biomolecular interaction, in particular enzymatic activity. Our calculus is based on a set of rewrite rules whose application depends on a probability. Each rule describes a transformation of one or more biomolecules. An interpretation algorithm and a formal semantics are given for the calculus and proved to be equivalent. We developed a prototype implementation of the interpreter for the calculus, which permits to follow the evolution of a biomolecular system. The interpreter is written in SICStus Prolog [20]. The formal semantics, given as a transition system, permits to verify properties of a system by model checking.

We used our calculus to study a real case of enzymatic activity, namely the reactions in the calf eye due to enzyme Sorbitol Dehydrogenase [15]. We show some results both of the interpreter and of the PRISM model checker [18].

Note that some of the formalisms proposed in [4, 7, 6, 8, 9, 12, 13, 16, 17, 19], use probability to model kinetic constants in reactions. Our calculus uses probability to model both kinetic constants and the effect of relative concentrations of reacting components.

The use of rules with a probability is not new. Probabilistic grammars and their applications have been considered by several authors (see as an example [10]).

2 Modelling Molecular Systems

We now define the syntax of solutions. We assume an infinite set \mathcal{E} of atomic particles.

Definition 1 (Solutions) Molecules m and solutions S are given by the following grammar:

$$\begin{array}{l} m ::= X \quad | \quad m-m \\ S ::= \mathbf{0} \quad | \quad m \quad | \quad S, S \end{array}$$

where X is any particle of \mathcal{E} , $\mathbf{0}$ stands for the empty solution, $m-m$ represents the complexation of two molecules, and S, S represents the union of two solutions.

With $\#(m)$ we denote the length of the molecule m , for example, if $m = A_1 - \dots - A_k$, where $A_1, \dots, A_k \in \mathcal{E}$, then $\#(m) = k$. We denote with \mathcal{E}^* the infinite set of molecules. With \mathcal{E}^n we denote the set of molecules of length at most equal to n , namely the set $\{m \in \mathcal{E}^* \mid \#(m) \leq n\}$. We remark that $\forall n > 0$, $\mathcal{E} \subseteq \mathcal{E}^n \subset \mathcal{E}^*$. We define the *concentration* of a molecule m in a solution S (denoted with $[m]_S$) as the number of occurrences of m in S . Finally, we denote with $\#(S)$ the *density* of the solution S . Intuitively, the density of a solution S represents the number of atomic particles that compose the solution. Formally, if $S = \mathbf{0}$ we have that $\#(S) = 0$, while if $S = m_1, m_2, \dots, m_k$ we have that $\#(S) = \sum_{i=1}^k \#(m_i)$.

Definition 2 (Structural congruence) The structural congruence for solutions \equiv is the smallest equivalence relation satisfying the following laws:

$$\begin{array}{l} S_1, S_2 \equiv S_2, S_1 \quad (S_1, S_2), S_3 \equiv S_1, (S_2, S_3) \quad S, \mathbf{0} \equiv S \\ m_1 - m_2 \equiv m_2 - m_1 \quad (m_1 - m_2) - m_3 \equiv m_1 - (m_2 - m_3) \end{array}$$

The syntax of solutions and the structural congruence relation allow us to consider a solution as a (possibly empty) multiset of molecules. For instance, the operator \in for solutions is defined as follows:

$$m \in S \iff \exists S' \text{ s.t. } m, S' \equiv S$$

In what follows we assume all operators for multisets to be defined also for solutions in a straightforward way.

Definition 3 (Probabilistic Rules) A probabilistic rule (or reaction) is a triple (S, P, S') where S and S' are solutions and P is a function from \mathcal{E}^* into $[0, 1[$.

Definition 4 (Well-formed set of rules) A set of rules $\mathcal{R} = (S_1, P_1, S'_1), \dots, (S_n, P_n, S'_n)$ is well-formed if it satisfies the following:

1. for all $(S_i, P_i, S'_i) \in \mathcal{R}$:
 - $S_i \cap S'_i = \emptyset$
 - $\#(S_i) = \#(S'_i)$
 - $P_i(m) > 0$ if $m \in S_i$, $P_i(m) = 0$ otherwise
 - if $m \equiv m'$ then $P_i(m) = P_i(m')$
2. for all $m \in \mathcal{E}^*$:
 - $P^I(m) = (1 - \sum_{i=0}^n P_i(m)) > 0$
3. if $(S_i, P_i, S'_i) \in \mathcal{R}$ then for all $j \in 1, \dots, n, j \neq i$: $S_i \not\equiv S_j$ or $S'_i \not\equiv S'_j$.

The value of $P^I(m)$ is said to be the probability of inaction of m . We denote the (not well-formed) rule (m, P^I, m) with R_m^I and the set $\{R_m^I \mid m \in \mathcal{E}^*\}$ with \mathcal{R}^I .

The intuition is that a rule (S, P, S') is applied to a solution containing S , in particular to a chosen S among the many possible ones contained in the solution, and that the rules to be applied are chosen with a probability. By condition 1 we require that in a reaction the overall number of atomic particles is preserved. Namely, molecules may form a complex, complexes may be split, but the overall number of atomic particles does not change.

As we shall say, the probability of the rule is a function of the probabilities of the molecules in the rule, and we require that the probability is greater than zero for the molecules in the left hand side of the rule and that occurrences of structurally congruent molecules have the same probability. Condition 2 expresses the requirement that the sum of the probabilities of all the possible transformations of a certain molecule is less than 1, leaving the possibility that the molecule is not transformed. This is expressed by the probability of inaction.

Condition 3 excludes rules that have both the same left hand side and the same right hand side, modulo structural congruence.

Example 1 Consider the following set of rules:

$$\mathcal{R} = \{R_1 = (\{A, B\}, P_1, \{A-B\}), R_2 = (\{A-B\}, P_2, \{A, B\})\}$$

where $P_1(A) = P_1(B) = \frac{2}{3}$ and $P_2(A-B) = \frac{1}{2}$. We have that the set of rules of inaction \mathcal{R}^I is as follows:

$$\begin{aligned} \mathcal{R}^I = & \{R_3 = (\{A\}, P_3, \{A\}), R_4 = (\{B\}, P_4, \{B\}), R_5 = (\{A-B\}, P_5, \{A-B\})\} \\ & \cup \{(\{X\}, P_X, \{X\}) \mid \text{for all } X \text{ in } \mathcal{E}^* \setminus \{A, B, A-B\}\} \end{aligned}$$

where $P^I(A) = P_3(A) = \frac{1}{3}$, $P^I(B) = P_4(B) = \frac{1}{3}$, $P^I(A-B) = P_5(A-B) = \frac{1}{2}$ and $P_X(X) = 1$ for all X . Therefore, the set of rules \mathcal{R} is well-formed. \square

Now we define (well-formed) molecular systems. In what follows we assume all sets of rules and all systems to be well-formed.

Definition 5 (System) *A system is a pair (S, \mathcal{R}) where S is a solution and $\mathcal{R} = \{(S_1, P_1, S'_1), \dots, (S_n, P_n, S'_n)\}$ is a finite set of probabilistic rules. A system (S, \mathcal{R}) is well-formed if \mathcal{R} is well-formed.*

3 A Semantics for Molecular Reactions

In this section we describe how molecular systems evolve over time. Roughly speaking, a system makes a sequence of steps which depend on the solution and on the probabilistic rules that can be applied to them. Each step may consist of the application of more than one rule in $\mathcal{R} \cup \mathcal{R}^I$. Each molecule must be involved in exactly one application of a rule. Therefore a solution behaves as a *fully parallel system* in which elements are able to perform reactions with a certain probability distribution. After a step we obtain a new solution.

Now we introduce a probabilistic algorithm, namely $Step(S)$, for describing how a system (S, \mathcal{R}) executes a step. The analysis of this algorithm allows us to compute the probability of S to be transformed into a new solution S' in a single step. We then use such a result in the definition of a probabilistic transition system for molecular reactions.

The algorithm $Step(S)$, shown in the picture below, is a recursive algorithm that (1) chooses an element in S , (2) applies a rule to it, possibly involving other elements, and (3) recursively executes on the rest of the solution. It is immediate to see that the algorithm is linear in the number of molecules of the solution $|S|$.

Algorithm 1 $Step(S)$

choose m in S

let R_1, \dots, R_n be the only rules in $\mathcal{R} \cup \mathcal{R}^I$
such that $R_i = (S_i, P_i, S'_i)$, $m \in S_i$, $S_i \subseteq S$

choose $R = (S_R, P_R, S'_R)$ in R_1, \dots, R_n
with probabilities $\frac{P_1(m)}{\sum_{i=1}^n P_i(m)}, \dots, \frac{P_n(m)}{\sum_{i=1}^n P_i(m)}$

if $S \setminus S_R = \emptyset$ then return S'_R
else return $(S'_R \cup Step(S \setminus S_R))$

In $Step(S)$ there are two probabilistic choices: the first selects a molecule m in the solution S and the second selects one of the probabilistic rules that can be applied to m . Since S is a multiset of molecules and since all instances of all molecules have the same probability of being chosen, the probability of choosing an instance of a molecule m is:

$$p(m) = \frac{[m]_S}{|S|}$$

Given m , one of the rules in $\mathcal{R} \cup \mathcal{R}^I$ has to be applied to it. The only rules that can be applied are the ones in which m appears in the left hand side, and such that all the elements in the left hand side are contained in the solution S . Let us denote these rules with R_1, \dots, R_n , the probability of $R_i = (S_i, P_i, S'_i)$ to be applied corresponds to $P_i(m)$ normalized to the sum of the probabilities of R_1, \dots, R_n , that is:

$$p(R_{i,1 \leq i \leq n} | m) = \frac{P_i(m)}{P_1(m) + \dots + P_n(m)}$$

Now we can compute the probability of S to be transformed into S' in one step, that is the probability of S' to be equal to the result of $Step(S)$. We can do this by calculating the probabilities of all the possible runs of $Step(S)$, then by discarding the runs that lead to something different from S' and finally by summing up the probabilities of the remaining ones.

Proposition 6 (Probability of a step) *The probability of $S' = Step(S)$ in a system (S, \mathcal{R}) is as follows:*

$$p(S, S', \mathcal{R}) = \begin{cases} 1 & \text{if } S = \emptyset = S' \\ 0 & \text{if } \#(S) \neq \#(S') \\ \sum_{i=1}^k p(m_i) \sum_{j=1}^{n_i} p(R_{i,1 \leq i \leq n_i} | m_i) p(S \setminus S_j, S' \setminus S'_j, \mathcal{R}) & \text{otherwise} \end{cases}$$

where m_1, \dots, m_k are all the molecules in S and R_1, \dots, R_{n_i} are all the probabilistic rules that can be applied to m_i in S .

Example 2 We recall the set of rules \mathcal{R} given in Example 1. The probability $p(\{A, B\}, \{A-B\}, \mathcal{R})$ of a solution A, B to be transformed in one step into $A-B$ is computed as follows:

$$\begin{aligned} p(\{A, B\}, \{A-B\}, \mathcal{R}) = & p(A)(p(R_1 | A)p(\emptyset, \emptyset, \mathcal{R}) + p(R_3 | A)p(\{B\}, \{A-B\} \setminus \{A\}, \mathcal{R})) \\ & + p(B)(p(R_1 | B)p(\emptyset, \emptyset, \mathcal{R}) + p(R_4 | B)p(\{A\}, \{A-B\} \setminus \{B\}, \mathcal{R})) \end{aligned}$$

where it is easy to check that $p(\{A\}, \{A-B\}, \mathcal{R}) = p(\{B\}, \{A-B\}, \mathcal{R}) = 0$, therefore:

$$p(\{A, B\}, \{A-B\}, \mathcal{R}) = \frac{1}{2} \cdot \left(\frac{2}{3} \cdot 1 + \frac{1}{3} \cdot 0 \right) + \frac{1}{2} \cdot \left(\frac{2}{3} \cdot 1 + \frac{1}{3} \cdot 0 \right) = \frac{2}{3}$$

and similarly we compute $p(\{A, B\}, \{A, B\}, \mathcal{R}) = \frac{1}{3}$, $p(\{A-B\}, \{A, B\}, \mathcal{R}) = \frac{1}{2}$ and $p(\{A-B\}, \{A-B\}, \mathcal{R}) = \frac{1}{2}$. \square

Given a system (S, \mathcal{R}) , let l be the density of S .

Definition 7 *Given a system (S, \mathcal{R}) , the Molecular Probabilistic Transition System (MPTS) of the system is a tuple $M = (Q^l, S, \mathcal{R}, \delta, \pi)$ where:*

- Q^l is a finite set of states, namely the set of solutions of density equal to l composed by molecules in \mathcal{E}^l ; $S \in Q^l$ is the initial state.

- $\delta \subseteq Q \times Q$ is a finite set of transitions.
- $\pi : \delta \rightarrow [0, 1]$ is a probability function that returns the probability associated with a transition. Namely, given a transition $e = (S, S')$, $\pi(e) = p(S, S', \mathcal{R})$.

Since the initial solution consists of a finite number of molecules, the set of rules in a system is finite and rules are well-formed, such that in particular $\#(S_i) = \#(S'_i)$, the number of states of the MPTS is finite.

Assuming that there is a transition $e = (S, S') \in \delta$ for each (S, S') such that $p(S, S', \mathcal{R}) > 0$ we have that $\forall S \in Q \sum_{(S, \bar{S}) \in \delta} \pi(S, \bar{S}) = 1$.

Example 3 We recall the set of rules \mathcal{R} of Example 1 and the probabilities computed in Example 2. The MPTS $M = (Q^l, S, \mathcal{R}, \delta, \pi)$ of system $(\{A, B\}, \mathcal{R})$ is shown in Figure 1. \square

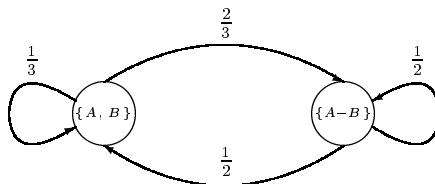


Fig. 1. Example of Molecular Probabilistic Transition System.

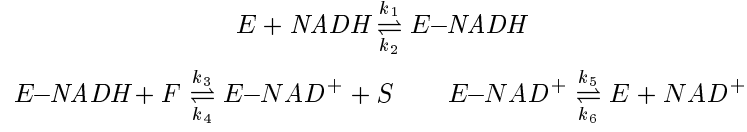
The following proposition states the correspondence between steps of the system, as computed by the algorithm, and transitions of the MPTS.

Proposition 8 (Correctness) *Given a system (S, \mathcal{R}) and its MPTS $M = (Q^l, S, \mathcal{R}, \delta, \pi)$, there is $e \in \delta$ such that $e = (S, S')$ with $\pi(e) = p(S, S', \mathcal{R})$ if and only if there is $S' = \text{Step}(S)$ with probability $p(S, S', \mathcal{R})$.*

4 An Application

In this section we report some experimental results of simulation of the activity of a molecular system and of model-checking its properties. We take as an example the reactions in the calf eye. Here the enzyme Sorbitol dehydrogenase (SDH) catalyzes the reversible oxidation of sorbitol and other polyalcohols to the corresponding keto-sugars. (The accumulation of sorbitol in the calf eye has been proposed as the primary event in the development of sugar cataract in the calf, see [15])

The reactions are depicted in the following scheme:



where E represents the enzyme Sorbitol dehydrogenase, S and F represent sorbitol and fructose, respectively, $NADH$ represents the nicotinamide adenine dinucleotide which stimulates the production of ATP in cells, and NAD^+ is the oxidized form of $NADH$; k_1, \dots, k_6 are the kinetic constants.

The enzyme E may react with $NADH$ and NAD^+ and give the complexes $E-NADH$ and $E-NAD^+$, respectively. The complex $E-NADH$ may react with a molecule of fructose F and give a complex $E-NAD^+$ and a molecule of sorbitol S . All the reactions are reversible and occur with different speeds according to their kinetic constants.

The above chain of reactions is expressed in our probabilistic rules as follows. Probabilities associated with each rule are chosen taking into account kinetic constants, given in [15].

Note that our calculus could be used to infer unknown kinetic constants by assuming different probabilities in the simulation of the considered system, until one does not obtain the expected results.

1. ($\{E, NADH\}, \{(E, 0.468), (NADH, 0.862)\}, \{E-NADH\}$)
2. ($\{E-NADH, F\}, \{(E-NADH, 0.001), (F, 0.002)\}, \{E-NAD^+, S\}$)
3. ($\{E-NAD^+\}, \{(E-NAD^+, 0.744)\}, \{E, NAD^+\}$)
4. ($\{E, NAD^+\}, \{(E, 0.457), (NAD^+, 0.859)\}, \{E-NAD^+\}$)
5. ($\{E-NAD^+, S\}, \{(E-NAD^+, 0.002), (S, 0.008)\}, \{E-NADH, F\}$)
6. ($\{E-NADH\}, \{(E-NADH, 0.772)\}, \{E, NADH\}$)
7. ($\{E\}, \{(E, 0.075)\}, \{E\}$)
8. ($\{S\}, \{(S, 0.992)\}, \{S\}$)
9. ($\{F\}, \{(F, 0.998)\}, \{F\}$)
10. ($\{NADH\}, \{(NADH, 0.138)\}, \{NADH\}$)
11. ($\{NAD^+\}, \{(NAD^+, 0.141)\}, \{NAD^+\}$)
12. ($\{E-NADH\}, \{(E-NADH, 0.227)\}, \{E-NADH\}$)
13. ($\{E-NAD^+\}, \{(E-NAD^+, 0.254)\}, \{E-NAD^+\}$)

Fig. 2. Probabilistic rules for Sorbitol dehydrogenase

4.1 Simulation

We implemented the interpreter using SICStus Prolog [20]. The input of the interpreter are a set of probabilistic rules describing the behaviour of the biological system and the initial solution.

E	S	F	$NADH$	NAD^+	$E-NADH$	$E-NAD^+$
10	100	100	100	100	0	0

Table 1. Initial solution

The interpreter gives the concentration of the molecules every millisecond. The initial solution used for the simulation experiment is given in Table 1.

The results of the interpreter are given in Figure 3. We see that, after an initial stabilization phase, the quantity of fructose is nearly four times the one of sorbitol. This agrees with the experimental results given in [15]. Simulations showed that different initial concentrations affect only the stabilization phase, as expected.

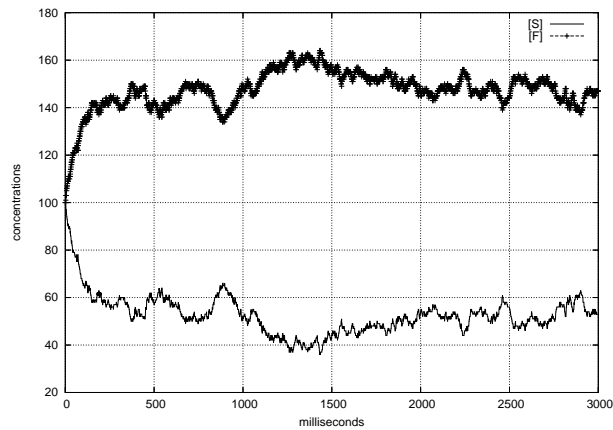


Fig. 3. Simulation of the activity of the Sorbitol Dehydrogenase: effects on the concentrations of Sorbitol and Fructose

4.2 Model Checking

Model checking is an automatic method for deciding whether a system satisfies a set of properties expressed in a probabilistic temporal logic. In [5], symbolic model checking is shown to be feasible for analyzing biological systems and advantageous over simulation when querying and validating formal models for biology. Some interesting queries one may consider when model checking the probabilistic evolutions of molecular systems are the following:

- Given a solution S , is there a pathway for synthesizing a molecule m ?

- From a solution S , is it possible to synthesize a molecule m without neither creating nor using a molecule m' ?
- Given a solution S , what is the probability for synthesizing a molecule m within a given amount of time?
- Can a molecule m reach a concentration greater than a given value τ ? With which probability?
- Which states may produce a concentration for m greater than τ ?
- Which states may produce a null concentration for m with a probability greater than a given value p ?
- Is a given relationship r between concentrations always satisfied?

The PRISM model checker

PRISM [18] is a probabilistic model checker that allows modeling and analyzing systems which exhibit a probabilistic behavior. Given a description of the system to be modelled, PRISM constructs a probabilistic model that can be either a *discrete-time Markov chain* (DTMC), a *Markov decision process* (MDP), or a *continuous-time Markov chain* (CTMC) [14]. On the constructed model PRISM can check properties specified by using a temporal logic (PCTL [11, 3, 2] for DTMCs and MDPs, and CSL [1] for CTMCs).

Since we used the model of CTMC for describing the molecular reactions in the bovine lens, we use the CSL specification language to specify properties of systems. The syntax of CSL is given by the following grammar:

$$\begin{aligned} \phi ::= & \text{true} \mid \text{false} \mid a \mid \phi \wedge \phi \mid \phi \vee \phi \mid \neg \phi \mid \mathcal{P}_{\sim p}[\psi] \mid \mathcal{S}_{\sim p}[\psi] \\ \psi ::= & X\phi \mid \phi \mathcal{U}^I \phi \mid \phi \mathcal{U} \phi \end{aligned}$$

where a is an atomic proposition, $\sim \in \{<, \leq, \geq, >\}$ is a relational operator, $p \in [0, 1]$ is a probability, and I is an interval of \mathbb{R} .

An atomic proposition a is satisfied or not by a given state of a Probabilistic Timed System. Symbol X denotes the “next state operator”, symbol \mathcal{U} denotes the “until” operator, and \mathcal{U}^I represents the “bounded until”. Intuitively, $\phi_1 \mathcal{U} \phi_2$ is satisfied when the formula ϕ_1 holds until ϕ_2 holds; $\phi_1 \mathcal{U}^I \phi_2$ is satisfied if ϕ_2 becomes true at some time instant t within the interval I ($t \in I$). Formula $\mathcal{P}_{\sim p}[\psi]$ is satisfied by a given set of computations iff the overall probability p' of the computations satisfying a path formula ψ is such that $p' \sim p$. Formula $\mathcal{S}_{\sim p}[\psi]$ is satisfied by a given set of states iff the overall probability p' of the states satisfying a state formula ψ is such that $p' \sim p$. Finally, we notice that the PRISM model checker permits to know the actual probability that a certain behaviour is observed by defining properties in the following way:

$$\begin{aligned} \mathcal{P}_{=?}[\psi] \\ \mathcal{S}_{=?}[\psi] \end{aligned}$$

where $[\psi]$ is either a path formula for \mathcal{P} or a state formula for \mathcal{S} .

Model Checking the Bovine Lens

In this section we show the results obtained verifying some properties on a PRISM specification of the molecular reactions arising in the bovine lens. Therefore, we translated the MPTS obtained according to the probabilities in Figure 2, into a PRISM specification. In particular, having fixed the initial concentrations of the molecules involved in the reactions in the bovine lens (thus obtaining an initial solution for the MPTS), and having fixed a time instant T , we studied the probability that, at time T , the concentration of S was greater than the concentrations of F . Moreover, we studied the probability that at instant T the concentration of S is smaller or equal than a certain value. Finally, we studied the probability that at instant T the concentration of the enzyme E is null.

S	F	NADH	NAD ⁺	E-NADH	E-NAD ⁺
25	25	25	25	0	0

Table 2. Initial solution

In Table 2 we show the concentrations of molecules in the initial solutions. We studied the probabilities mentioned above, according to different values of the initial concentration of E .

As we have seen, the molecular reactions in the bovine lens transform molecules of S and NAD^+ into molecules of F and $NADH$. In Figure 4 we show the probability that the concentration of S becomes less than 10 (namely, at least 15 molecules of S are transformed into F). In particular, we checked the property:

$$\mathcal{P}_{=?}[true \mathcal{U}^{\leq T} [S] \leq 10]$$

where T is fixed at value 5000 which corresponds in the real model at about 500 milliseconds. On the axis of such graphs we put the initial concentration of enzyme E .

Moreover, we would like to investigate with which probabilities anomalous cases may arise (for example, as said above, we study the probability to get a solution where the concentration of S is greater than the concentration of F). In particular we checked the property:

$$\mathcal{P}_{=?}[true \mathcal{U}^{\leq T} [S] > [F]]$$

where T is fixed at value 5000 and the initial concentration of E fixed at value 5, and we got a probability of $4.818E - 6$ (showing that such an anomalous case arises with an extremely low probability).

Finally, Figure 5 shows the probability that the concentrations of E reaches a value 0 by varying the initial concentration of E . The checked property is:

$$\mathcal{P}_{=?}[true \mathcal{U}^{\leq T} [E] = 0]$$

where, again, T is fixed at value 5000 and on the axis of the graph we put the initial concentration of enzyme E .

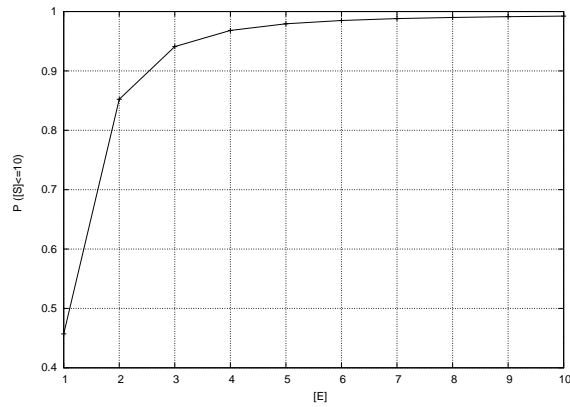


Fig. 4. $\mathcal{P}_{=?}[\text{true } U^{\leq T} [S] \leq 10]$

5 Conclusions

We have introduced a calculus for molecular reactions and demonstrated its use for studying a real biological system. We implemented an interpreter for simulating reactions and used a model checker for verifying their properties. An extensive experimentation will allow tuning the model with the longtime aim of offering a working tool to biologists.

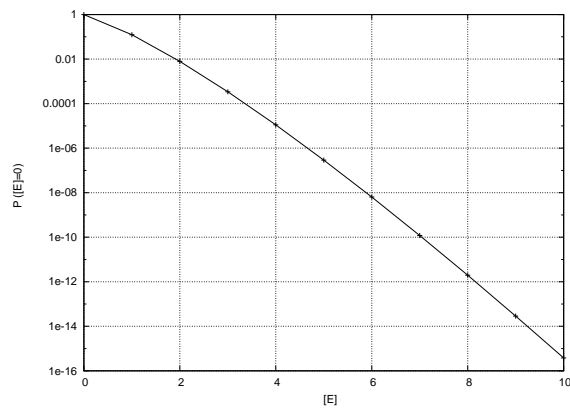


Fig. 5. $P([E] = 0)$

References

1. C. Baier, B. Havercort, H. Hermanns, J. P. Kaoten. Model Checking Continuous-time Markov Chains by Transient Analysis. Proc. of CAV'00, LNCS 1855, 358–372, 2000.
2. C. Baier, M. Kwiatowska. Model Checking for a Probabilistic Branching Time Logic with Fairness. *Distributed Computing* 11(3), 125–155, 1998.
3. A. Bianco, L. de Alfaro. Model Checking of Probabilistic and Nondeterministic Systems. Proc. Int. Conference on Foundation of Software Technologies and Theoretical Computer Science, LNCS 1026, 499–513, 1995.
4. L. Cardelli. Bioware Languages. In A. Herbert and K. Spaark eds, Computer systems: Papers for Roger Needham, to appear.
5. N. Chabrier, F. Fages. Symbolic Model Checking of Biochemical Networks. Proc. of CSMB'03, LNCS 2602, 149–162, 2003.
6. N. Chabrier-Rivier, F. Fages, and S. Soliman. The Biochemical Abstract Machine BIOCHAM. Computational Methods in Systems Biology 2004, LNCS, 2004, to appear.
7. M. Curti, P. Degano, C.T. Baldari. Causal pi-Calculus for Biochemical Modelling. Computational Methods in Systems Biology, LNCS 2602, 2003. 21–33.
8. V. Danos, C. Laneve. Core Formal Molecular Biology. European Symposium on Programming, LNCS 2618, 302–318, 2002.
9. V. Danos, C. Laneve. Graphs for Core Molecular Biology. Computational Methods in Systems Biology, LNCS 2602, 34–46, 2003.
10. S. Geman, M. Johnson. Probabilistic Grammars and their Applications. In N.J. Smelser, P.B. Baltes eds *International Encyclopedia of the Social & Behavioral Sciences*, Pergamon, Oxford, 12075-12082, 2002.
11. H. Hansson, B. Jonsson. A Logic for Reasoning About Time and Probability. *Formal Aspects of Computing* 6(5), 512–535, 1994.
12. D. Harel. A Grand Challenge: Full Reactive Modeling of a Multi-cellular Animal. Hybrid Systems: Computation and Control, LNCS 2623, 2003.
13. N. Kam, D. Harel, H. Kugler, R. Marelly, A. Pnueli, E.J.A. Hubbard, M.J. Stern. Formal Modeling of *C. elegans* Development: A Scenario-Based Approach. Computational Methods in Systems Biology, LNCS 2602, 4–20, 2003.
14. M. Kwiatkowska. Model Checking for Probability and Time: From Theory to Practice. Proc. of LICS'03, IEEE CS Press, 351–360, 2003.
15. I. Marini, L. Bucchioni, P. Borella, A. Del Corso, and U. Mura. Sorbitol Dehydrogenase from Bovine Lens: Purification and Properties. *Archives of Biochemistry and Biophysics*, 340, 383–391, 1997.
16. M. Nagasaki, S. Onami, S. Miyano, and H. Kitano. Bio-calculus: Its Concept and Molecular Interaction. *Genome Informatics* 10, 133–143, 1999.
17. C. Priami, A. Regev, E.Y. Shapiro, W. Silverman. Application of a Stochastic Name-passing Calculus to Representation and Simulation of Molecular Processes. *Information Processing Letters* 80, 25–31, 2001.
18. PRISM URL: <http://www.cs.bham.ac.uk/~dxdp/prism>.
19. A. Regev, W. Silverman, and E.Y. Shapiro. Representation and Simulation of Biochemical Processes Using the pi-Calculus Process Algebra. Pacific Symposium on Biocomputing 2001, 459–470, 2001.
20. SICStus URL: <http://www.sics.se/sicstus/>.