A Probabilistic Model for Molecular Systems

Roberto Barbuti

Stefano Cataudella

Andrea Maggiolo–Schettini

Paolo Milazzo [C]

Angelo Troina

Dipartimento di Informatica, Università di Pisa

Largo B. Pontecorvo 3, 56127 - Pisa, Italy

E-mail: {*barbuti,cataudel,maggiolo,milazzo,troina*}*@di.unipi.it*

Abstract. We introduce a model for molecular reactions based on probabilistic rewriting rules. We give a probabilistic algorithm for rule applications as a semantics for the model, and we show how a probabilistic transition system can be derived from it. We use the algorithm in the development of an interpreter for the model, which we use to simulate the evolution of molecular systems. In particular, we show the results of the simulation of a real example of enzymatic activity. Moreover, we apply the probabilistic model checker PRISM to the transition system derived by the model of this example, and we show the results of model checking of some illustrative properties.

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. Among the many formalisms that have been applied to biology there are, for instance, Petri Nets [19, 17], Hybrid Systems [1], and the π–calculus [23, 8]. In particular, a stochastic variant of the latter, based on the algorithm given by Gillespie in [11], has been used for the development of a simulator for molecular systems [20, 22]. Moreover,

[C]Corresponding author

models for the design of reactive systems have been applied to multicellular organisms [14, 13] and some new formalisms have been proposed to describe biomolecular and membrane interactions [5, 7, 9, 18].

In this paper we introduce a probabilistic model for biomolecular interaction which is based on a set of rewrite rules whose application depends on a probability. Each rule describes a reaction, which is a transformation of one or more biomolecules. We give an interpretation algorithm and a formal semantics for the model and we prove their equivalence. We developed a prototype implementation of the interpreter for the model, which permits to follow the evolution of a biomolecular system. The interpreter is written in SICStus Prolog [24]. The formal semantics, given as a transition system, allows verifying properties of a system by model checking.

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]). A stochastic calculus has been proposed in [20] for describing molecular processes. The novelty of our approach is the use of probabilities for modelling both the effects of concentration of reactants and speed of reactions.

We used our model to study a real case of enzymatic activity, namely the reactions in the calf eye due to enzyme Sorbitol Dehydrogenase [16]. We show some results both of the interpreter and of the PRISM model checker [21].

In Section 2, we introduce the model and we define the formal syntax of a biochemical system. In Section 3, we define a step semantics for the model by giving a probabilistic algorithm to compute the step of the system. Moreover, we show how this algorithm allows the derivation of a probabilistic transition system which describes the possible evolutions of a biochemical system. In Section 4, we show the results of simulation and model checking of the Sorbitol Dehydrogenase activity and, finally, in Section 5, we give some conclusions.

2. Modelling Molecular Systems

With the term *molecular system* we denote a chemical solution in which some reactions may occur. A chemical solution contains *molecules* that are compositions of one or more *elementary particles*. The definition of elementary particle may depend on the level of detail used to describe reactions: for instance, for a chemist elementary particles can be atoms, while for a biologist they can be proteins and enzymes. Hence, for the sake of generality, in the definition of molecules we assume an infinite set $\mathcal{E}$ of elementary particles, without giving details of what they correspond to in the real world.

We now give the formal definition of molecules and solutions.

Definition 2.1. (Solutions)
Molecules m and *solutions* S are given by the following grammar:

$$\begin{array}{rcl} m & ::= & X \;\big|\; m\text{–}m \\ S & ::= & \mathbf{0} \;\big|\; m \;\big|\; S, S \end{array}$$

where X is any particle of $\mathcal{E}$, **0** stands for the empty solution, m–m represents the complexation of two molecules, and S, S represents the union of two solutions. With $\mathcal{S}$ we denote the set of all the possible solutions.

With $\#(m)$ we denote the length of the molecule m, for example, if $m = X_1 - \ldots - X_k$, where $X_1, \ldots X_k \in \mathcal{E}$, 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 elementary particles that compose the solution. Formally, if $S = \mathbf{0}$ we have that $\#(S) = 0$, while if $S = m_1, m_2, \ldots, m_k$ we have that $\#(S) = \sum_{i=1}^{k} \#(m_i)$.

Definition 2.2. (Structural congruence)
The *structural congruence* for solutions $\equiv$ is the smallest equivalence relation satisfying the following laws:

$$S_1, S_2 \equiv S_2, S_1 \qquad (S_1, S_2), S_3 \equiv S_1, (S_2, S_3) \qquad S, \mathbf{0} \equiv S$$
$$m_1 – m_2 \equiv m_2 – m_1 \qquad (m_1 – m_2) – m_3 \equiv m_1 – (m_2 – m_3)$$

The structural congruence allows us to consider the equivalence classes induced by such a relation as (possibly empty) multisets. For instance, let $S_\equiv$ be the multiset representing the equivalence class of the solution S; the operator $\in$ can be defined as follows:

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

Moreover, union and intersection are as follows:

$$S'_\equiv \cup S''_\equiv = (S', S'')_\equiv$$

$$S'_\equiv \cap S''_\equiv = S_\equiv \quad \textit{where } [m]_S = \begin{cases} min([m]_{S'}, [m]_{S''}) & \textit{if } m \in S' \textit{ and } m \in S'' \\ 0 & otherwise \end{cases}$$

Other operators and relations such as $\setminus$ and $\subset$ can be easily defined in terms of $\cup$ and $\cap$. In what follows we will use all these operator also with solutions. For instance, with $S' \cup S''$ we will mean *one of the solutions in the equivalence class* $S'_\equiv \cup S''_\equiv$. Formally, $S = S' \cup S''$ if and only if $S_\equiv = S'_\equiv \cup S''_\equiv$.

Now we describe how chemical reactions may occur in a solution. A chemical reaction is represented by a *probabilistic rewrite rule* transforming some molecules of a solution into some others. The probability of the rule will be used to model the speed of the reaction: the faster is the chemical reaction, the higher should be the probability of the corresponding rule. In Section 3 we will describe how probabilities are used in the application of the rules and we will show that also concentrations of molecules play an important role.

Definition 2.3. (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 $\left[0, 1\right[$.

In order to describe all the reactions that may occur in a solution, we consider a *set of rules*. We require that rules of the set satisfy some constraints of well–formedness. These constraints will allow us to define the semantics of the calculus and to guarantee the finiteness of the states (different solutions) that can be reached by applying rules in accordance with the semantics.

Definition 2.4. (Well–formed set of rules)
A set of rules $\mathcal{R} = \{(S_1, P_1, S'_1), \ldots \ldots, (S_n, P_n, S'_n)\}$ is well–formed if it satisfies the following conditions:

1. for all $(S_i, P_i, S'_i) \in \mathcal{R}$:
   (a) $S_i \cap S'_i = \varnothing$
   (b) $\#(S_i) = \#(S'_i)$
   (c) $P_i(m) > 0$ if $m \in S_i$, $P_i(m) = 0$ otherwise
   (d) if $m \equiv m'$ then $P_i(m) = P_i(m')$
2. for all $m \in \mathcal{E}^*$:
   $$P^I(m) = (1 - \textstyle\sum_{i=0}^{n} P_i(m)) > 0$$
3. if $(S_i, P_i, S'_i) \in \mathcal{R}$ then for all $j \in 1, \ldots, n$ if $S_i \equiv S_j$ and $S'_i \equiv S'_j$ then $i = j$.

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^I_m and the infinite set $\{R^I_m \mid m \in \mathcal{E}^*\}$ 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 there are no molecules in the left hand side of a rule that are left unchanged after application (a), and that the overall number of elementary particles is preserved (b). Namely, molecules may form complexes and complexes may be split, but without changing the overall number of elementary particles.

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 it is zero otherwise (c), and that occurrences of structurally congruent molecules have the same probability (d). 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 (but possibly different probabilities), modulo structural congruence.

Example 2.1. 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 in $\mathcal{E}^* \setminus \{A, B, A–B\}$. Therefore, the set of rules $\mathcal{R}$ is well–formed.

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

Definition 2.5. (System)
A *molecular system* is a pair $(S, \mathcal{R})$, where S is a solution and $\mathcal{R} = \{(S_1, P_1, S'_1), \ldots, (S_n, P_n, S'_n)\}$ is a finite set of probabilistic rules. A molecular 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 performs a sequence of steps which depend on the solutions 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 this result in the definition of a probabilistic transition system for molecular reactions.

The algorithm $Step(S)$, shown 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 complexity of the algorithm is linear in the number of molecules of the solution $|\,S\,|$.

Algorithm 1 $Step(S)$

```
choose $m$ in S
let $R_1, \ldots, R_n$ 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 $R_j = (S_j, P_j, S'_j)$ in $R_1, \ldots, R_n$ with probability $\frac{P_j(m)}{\sum_{i=1}^{n} P_i(m)}$
if $S \setminus S_j = \varnothing$ then return $S'_j$ else return ($S'_j \cup Step(S \setminus S_j)$)
```

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 all instances of all molecules in S have the same probability of being chosen, the probability of choosing an instance of a molecule m is:

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

The probability $p(m)$ reflects the concentration of the molecule m in 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. We denote these rules with $R_1, \ldots, R_n$; the probability to apply

$R_j = (S_j, P_j, S'_j)$, for $1 \leq j \leq n$, corresponds to $P_j(m)$ normalized to the sum of the probabilities of $R_1, \ldots, R_n$, that is:

$$p(R_j \mid m) = \frac{P_j(m)}{P_1(m) + \ldots + P_n(m)}$$

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 $Step(S)$. This is done 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.

Definition 3.1. (Probability of a step)
The probability of $S' = Step(S)$ in a system $(S, \mathcal{R})$ is as follows:

$$p(S, S', \mathcal{R}) = \begin{cases} 1 & \textit{if } S = \varnothing = S' \\ 0 & \textit{if } \#(S) \neq \#(S') \\ \sum_{i=1}^{k} p(m_i) \sum_{j=1}^{n_i} p(R_j \mid m_i) p(S \backslash S_j, S' \backslash S'_j, \mathcal{R}) & \textit{otherwise} \end{cases}$$

where $m_1, \ldots, m_k$ are all the molecules in S and $R_1, \ldots, R_{n_i}$, where $R_j = (S_j, P_j, S'_j)$, are all the probabilistic rules that can be applied to m_i in S.

We now give a couple of propositions about the probability of a step. The first proposition implies that the probability of performing a step from a solution in a certain equivalence class to another solution in another class depends only on the classes of the solutions. The second proposition implies that the probability of a step is a well–defined probability distribution over equivalence classes.

Proposition 3.1. Given a well–formed set of rules $\mathcal{R}$ and solutions S_1, S_2, S'_1, S'_2 such that $S_1 \equiv S_2$ and $S'_1 \equiv S'_2$, it holds $p(S_1, S'_1, \mathcal{R}) = p(S_2, S'_2, \mathcal{R})$.

Proof:
Follows directly from the definition of $p(S, S', \mathcal{R})$ and by the well–formedness of $\mathcal{R}$. □

Proposition 3.2. Given a well–formed set of rules $\mathcal{R}$ and a solution S, it holds:

$$\sum_{S'_{\equiv} \subset \mathcal{S}} p(S, S', \mathcal{R}) = 1$$

Proof:
We prove the proposition by induction on the number of molecules in S.

The definitions of $p(m)$ and of $p(R_i \mid m)$ guarantee that $\sum_{i=1}^{k} p(m_i) = 1$ and $\sum_{j=1}^{n_i} p(R_j \mid m_i) = 1$. It is easy to verify that the proposition holds for $\mid S \mid = 0$ and $\mid S \mid = 1$. Assuming by induction hypothesis that, given an index j, $\sum_{S'_{\equiv} \subset \mathcal{S}} p(S \setminus S_j, S' \setminus S_j, \mathcal{R}) = 1$, we have that $\sum_{S'_{\equiv} \subset \mathcal{S}} p(S, S', \mathcal{R}) = 1$. □

Example 3.1. We recall the set of rules $\mathcal{R}$ given in Example 2.1. 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:

$$\begin{aligned} p(\{A, B\}, \{A–B\}, \mathcal{R}) = \\ & p(A)\big(p(R_1 \mid A)p(\varnothing, \varnothing, \mathcal{R}) + p(R_3 \mid A)p(\{B\}, \{A–B\} \setminus \{A\}, \mathcal{R})\big) \\ & \quad + p(B)\big(p(R_1 \mid B)p(\varnothing, \varnothing, \mathcal{R}) + p(R_4 \mid B)p(\{A\}, \{A–B\} \setminus \{B\}, \mathcal{R})\big) \end{aligned}$$

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

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

and similarly it is possible to compute $p(\{A,B\},\{A,B\},\mathcal{R}) = \frac{1}{3}$, $p(\{A–B\},\{A,B\},\mathcal{R}) = \frac{1}{2}$ and $p(\{A–B\},\{A–B\},\mathcal{R}) = \frac{1}{2}$.

Now, in order to verify properties of molecular systems we define the *Molecular Probabilistic Transition System (MPTS)*, that is a probabilistic transition system in which each state corresponds to an equivalence class of solutions and each transition corresponds to a possible application of the $Step$ algorithm for a given system $(S,\mathcal{R})$.

Definition 3.2. (Molecular Probabilistic Transition System)
Given a system $(S,\mathcal{R})$, the *Molecular Probabilistic Transition System* (MPTS) of the system is a tuple $M = (Q, S_\equiv, \mathcal{R}, \delta, \pi)$ where:

- Q is a set of states, namely the set of equivalence classes of solutions composed by molecules in $\mathcal{E}^*$; $S_\equiv \in Q$ is the initial state.
- $\delta \subseteq Q \times Q$ is a set of transitions.
- $\pi : \delta \rightarrow [0,1]$ is a probability function that returns the probability associated with a transition. Namely, given a transition $e = (S_\equiv, S'_\equiv), \pi(e) = p(S,S',\mathcal{R})$.

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

Proposition 3.3. (Correctness)
Given a system $(S,\mathcal{R})$ and the corresponding MPTS $M = (Q, S_\equiv, \mathcal{R}, \delta, \pi)$, there exists $e \in \delta$ such that $e = (S_\equiv, S'_\equiv)$ with $\pi(e) = p(S,S',\mathcal{R})$ if and only if there is $S' = Step(S)$ with probability $p(S,S',\mathcal{R})$. Moreover, it holds $\forall S \in Q$, $\sum_{(S_\equiv,S'_\equiv)\in\delta} \pi(S_\equiv,S'_\equiv) = 1$.

Proof:
By the construction of the MPTS and by Propositions 3.1 and 3.2. □

A *path* from a state $S_\equiv$ to a state $S'_\equiv$ in a MPTS M is a sequence of transitions $\delta_0,\ldots,\delta_n$ such that $\delta_0 = (S_\equiv, S^1_\equiv),\ldots,\delta_i = (S^i_\equiv, S^{i+1}_\equiv),\ldots,\delta_n = (S^n_\equiv, S'_\equiv)$ for some states $S^1_\equiv,\ldots,S^n_\equiv$. The probability of a path is the product of the probabilities of its transitions. We say that a state of a MPTS is *reachable* if there exists a path form the initial state to such state having probability greater than zero.

The well-formedness of molecular systems allows us to prove that the number of reachable states in a MPTS is finite.

Proposition 3.4. (Finiteness)
Given a well-formed molecular system $(S,\mathcal{R})$, the number of reachable states in the corresponding MPTS $M = (Q, S_\equiv, \mathcal{R}, \delta, \pi)$ is finite.

Proof:
If $\mathcal{R} = \varnothing$, then $p(S, S', \mathcal{R}) = 1$ if $S \equiv S'$, and $p(S, S', \mathcal{R}) = 0$ otherwise. Hence, the only reachable state in M is $S_{\equiv}$.

If $\mathcal{R} = \{R_1, \ldots, R_n\}$ with $n \geq 1$, then the constraint $\#(S_i) = \#(S'_i)$ of well-formedness guarantees that all the reachable states correspond to classes of solutions with density $\#(S)$. Moreover, the finiteness of the set of rules and of the size of the initial solution implies that the number of elementary particles that can occur during the evolution is finite too. Hence, we have that the possible states are at most as many as the possible classes of solutions with density $\#(S)$ and containing at most finitely many different elementary particles, thus also the possible states are finitely many. □

Example 3.2. We recall the set of rules $\mathcal{R}$ of Example 2.1 and the probabilities computed in Example 3.1. The MPTS $M = (Q, \{A, B\}_{\equiv}, \mathcal{R}, \delta, \pi)$ of system $(\{A, B\}, \mathcal{R})$ is shown in Figure 1.

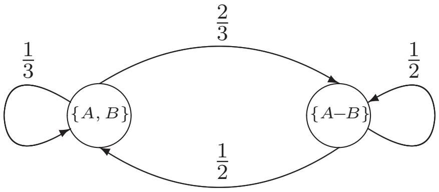

Figure 1. Example of Molecular Probabilistic Transition System.

4. An Application

In this section we report on some experimental results about the simulation of the activity of a molecular system and of model-checking its properties.

We take as an example some 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 [16]). The reactions are shown in the following scheme:

$$E + \mathit{NADH} \underset{k_2}{\overset{k_1}{\rightleftharpoons}} E\text{–}\mathit{NADH}$$

$$E\text{–}\mathit{NADH} + F \underset{k_4}{\overset{k_3}{\rightleftharpoons}} E\text{–}NAD^+ + S \qquad E\text{–}NAD^+ \underset{k_6}{\overset{k_5}{\rightleftharpoons}} E + NAD^+$$

where E represents the enzyme Sorbitol Dehydrogenase, S and F represent sorbitol and fructose, respectively, *NADH* represents the nicotinamide adenine dinucleotide, and NAD^+ is the oxidized form of $NADH$; $k_1, \ldots, k_6$ are the kinetic constants.

The enzyme E may react with *NADH* and NAD^+ giving the complexes E–*NADH* and $E\text{–}NAD^+$, respectively. Moreover, the *E–NADH* complex may react with a molecule of fructose F giving a complex $E\text{–}NAD^+$ and a molecule of sorbitol S. All the reactions are reversible and occur with different speeds according to their kinetic constants.

1. $(\{E, \mathit{NADH}\}, \{(E, 0.0054), (\mathit{NADH}, 0.0062)\}, \{E\text{–}\mathit{NADH}\})$
2. $(\{E\text{–}\mathit{NADH}, F\}, \{(E\text{–}\mathit{NADH}, 0.000002), (F, 0.000002)\}, \{E\text{–}NAD^+, S\})$
3. $(\{E\text{–}NAD^+\}, \{(E\text{–}NAD^+, 0.227)\}, \{E, NAD^+\})$
4. $(\{E, NAD^+\}, \{(E, 0.00057), (NAD^+, 0.00061)\}, \{E\text{–}NAD^+\})$
5. $(\{E\text{–}NAD^+, S\}, \{(E\text{–}NAD^+, 0.000007), (S, 0.000008)\}, \{E\text{–}\mathit{NADH}, F\})$
6. $(\{E\text{–}\mathit{NADH}\}, \{(E\text{–}\mathit{NADH}, 0.033)\}, \{E, \mathit{NADH}\})$
7. $(\{E\}, \{(E, 0.99403)\}, \{E\})$
8. $(\{S\}, \{(S, 0.999992)\}, \{S\})$
9. $(\{F\}, \{(F, 0.999998)\}, \{F\})$
10. $(\{\mathit{NADH}\}, \{(\mathit{NADH}, 0.999998)\}, \{\mathit{NADH}\})$
11. $(\{NAD^+\}, \{(NAD^+, 0.99939)\}, \{NAD^+\})$
12. $(\{E\text{–}\mathit{NADH}\}, \{(E\text{–}\mathit{NADH}, 0.966998)\}, \{E\text{–}\mathit{NADH}\})$
13. $(\{E\text{–}NAD^+\}, \{(E\text{–}NAD^+, 0.772993)\}, \{E\text{–}NAD^+\})$

Figure 2. Probabilistic rules for Sorbitol Dehydrogenase

The above chain of reactions is expressed by the probabilistic rules in Figure 4. Probabilities associated with each rule are chosen taking into account kinetic constants, given in [16]. Note that our model 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.

4.1. Simulation

We implemented the interpreter using SICStus Prolog [24]. The input of the interpreter are an initial solution and a set of probabilistic rules describing the behaviour of the biological system. The interpreter gives the concentration of the molecules at each step. The initial solution used for the simulation experiment is given in Table 1. The results of the interpreter are shown in Figure 3. We see that, after an initial stabilization phase, the concentration of the sorbitol becomes nearly 10 times the one of the fructose.

E	S	F	*NADH*	NAD^+	*E–NADH*	$E\text{–}NAD^+$
5	100	100	100	100	0	0

Table 1. Initial solution for the simulation of the activity of the Sorbitol Dehydrogenase.

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 [6], symbolic model checking is shown to be feasible for

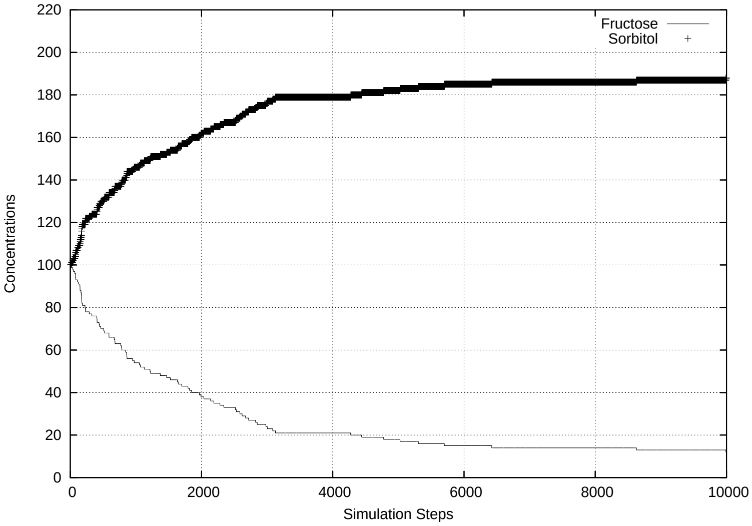

Figure 3. Simulation of the activity of the Sorbitol Dehydrogenase. The concentration of the sorbitol (thick line) becomes nearly 10 times greater than the concentration of the fructose (thin line)

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 p?
- Is a given relationship r between concentrations always satisfied?

The PRISM model checker

PRISM [21] is a probabilistic model checker that allows modelling and analyzing systems which exhibit a probabilistic behaviour. 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) [15]. On the constructed model PRISM can check properties specified by using a temporal logic (Probabilistic Computation Tree Logic (PCTL) [12, 4, 2] for DTMCs and MDPs, and Continuous Stochastic Logic (CSL) [3] for CTMCs).

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

$$\phi ::= true \mid false \mid a \mid \phi \wedge \phi \mid \phi \vee \phi \mid \neg\phi \mid \mathcal{P}_{\sim p}[\psi] \mid \mathcal{S}_{\sim p}[\psi]$$

$$\psi ::= X\phi \mid \phi\,\mathcal{U}^I\,\phi \mid \phi\,\mathcal{U}\,\phi$$

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 non–negative reals. Symbol X denotes the "next state operator", symbol $\mathcal{U}$ denotes the "until" operator, and $\mathcal{U}^I$ represents the "bounded until".

A formula ϕ expresses a property of a state (state fomula), while a formula ψ expresses a property of paths exiting a state (path formula). Intuitively, formula $\mathcal{P}_{\sim p}[\psi]$ is satisfied from a given state if and only if the overall probability p' of the computations satisfying a path formula ψ is such that $p' \sim p$. Formula $\mathcal{S}_{\sim p}[\psi]$ asserts that the steady–state probability p' of being in a state satisfying ψ is such that $p' \sim p$. The steady state probability is the probability of being in a certain state in the long run. Moreover, the path formula $\phi_1\mathcal{U}\phi_2$ is satisfied when ϕ_1 holds until ϕ_2 holds, and $\phi_1\mathcal{U}^I\phi_2$ is satisfied if ϕ_1 holds and then ϕ_2 becomes true at some time instant t within the interval I ($t \in I$).

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:

$$\mathcal{P}_{=?} \quad [\psi]$$
$$\mathcal{S}_{=?} \quad [\psi]$$

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 by verifying some illustrative 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), we studied the probability that, at a given time T, the concentration of S is greater than the concentrations of F, and vice–versa, that the concentration of F is greater than the concentration of S. Moreover, we studied the probability that after a few time the concentration of the enzyme E is null. 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 F and *NADH* into molecules of S and NAD^+. In Figure 4 we show the probability that the concentration of S becomes greater than the concentration of F and vice–versa, in the time interval $[0, T]$. In particular we checked the properties:

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

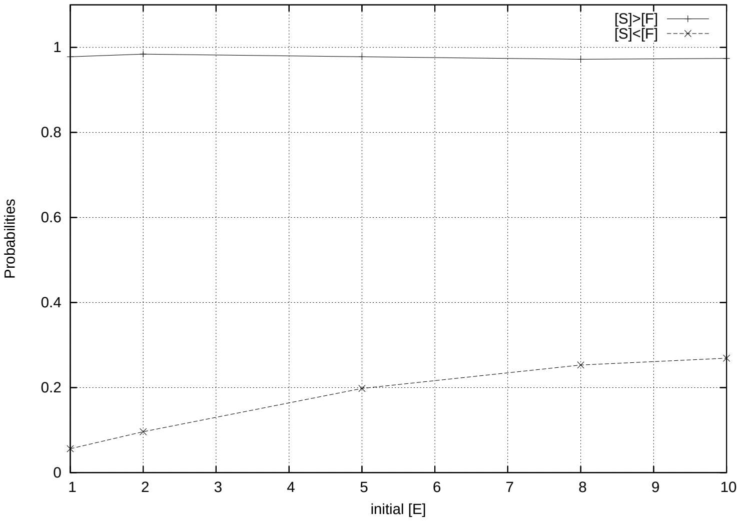

Figure 4. Probability that the concentration of the sobitol becomes greater than the concentration of the fructose and vice–versa, by varying the initial concentration of the enzyme.

S	F	*NADH*	NAD^+	*E–NADH*	*E–*NAD^+
25	25	25	25	0	0

Table 2. Concentrations of molecules in the initial solution used for model checking. Properties are checked by varying the initial concentration of the enzyme.

where T is fixed at 10000 and the initial concentration of E varies between 1 and 10. As expected, the probability that $[S] > [F]$ is greater than the probability that $[S] < [F]$, which confirms the results of simulation. Furthermore, the latter increases when the initial concentration of the enzyme increases. This happens because the activity of the enzyme is proportional to its concentration.

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

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

where T is fixed to the very small value 5. This property shows the affinity of the enzyme with NAD^+ and *NADH*, because $[E]$ is equal to zero when the enzyme is bound to them. The fact that the probability is close to one when the initial concentration is small demonstrates that the affinity is good. The probability decreases with greater initial values of $[E]$ because we considered a very small value for T, which corresponds to a very short initial time interval.

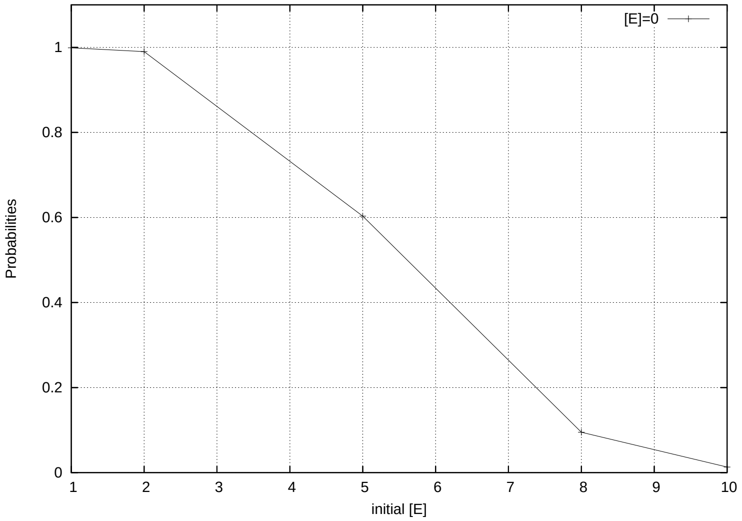

Figure 5. Probability that the concentrations of the enzyme reaches 0 by varying its initial value.

5. Conclusions

We have introduced a model for molecular reactions that deals with quantitative aspects of biochemical systems (speed of reactions) by using probabilities. Among the formalisms we have mentioned ([5, 7, 8, 9, 13, 14, 18, 19, 20, 23]) only [19] and [20] deal with quantitative aspects. In particular [20] uses probabilities to model speed of reactions, but with a different approach. In fact, in [20] the kinetic constants of the reactions and the concentrations of the reactants are used to compute rates, and rates are used to choose among possible reactions. Instead, we use concentration of reactants in order to choose one molecule of the solution, and we use probabilities that are related to kinetic constants in order to choose the reaction that has to occur.

We implemented an interpreter based on the model. We used it for simulating reactions of a real biological system and we 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.

Acknowledgments

This work has been inspired by many discussions with Antonella Del Corso and Alberto Mura of the Department of Physiology and Biochemistry of the University of Pisa. We thank also Pierpaolo Degano and Cosimo Laneve who gave us useful suggestions.

References

[1] R. Alur, C. Belta, F. Ivancic, V. Kumar, M. Mintz, G. Pappas, H. Rubin, and J. Schug. *Hybrid Systems : Computation and Control*, volume 2034 of *LNCS*, chapter Hybrid Modeling and Simulation of Biomolecular Networks, pages 19–32. Springer–Verlag, 2001.

[2] C. Baier and M. Kwiatowska. Model checking for a probabilistic branching time logic with fairness. *Distributed Computing*, 11(3):125–155, 1998.

[3] Christel Baier, Boudewijn R. Haverkort, Holger Hermanns, and Joost-Pieter Katoen. Model checking continuous-time markov chains by transient analysis. In *Proc. of CAV*, volume 1885 of *LNCS*, pages 358–372. Springer, 2000.

[4] A. Bianco and L. de Alfaro. Model checking of probabilistic and nondeterministic systems. In *Proc. of Int. Conference on Foundation of Software Technologies and Theoretical Computer Science*, volume 1026 of *LNCS*, pages 499–513, 1995.

[5] L. Cardelli. Bioware languages. In *Computer Systems. Theory, Technology and Applications. Papers for Roger Needham*, pages 59–66. Springer, 2003.

[6] N. Chabrier and F. Fages. Symbolic model checking of biochemical networks. In *Comp. Methods in Systems Biology (CMSB)*, volume 2602 of *LNCS*, pages 149–162, 2003.

[7] N. Chabrier-Rivier, F. Fages, and S. Soliman. The biochemical abstract machine biocham. In *Comp. Methods in Systems Biology (CMSB)*, 2004.

[8] M. Curti, P. Degano, C. Priami, and C.T. Baldari. Modelling biochemical pathways through enhanced π–calculus. *Theoretical Computer Science*, 325(1):111–140, 2004.

[9] Vincent Danos and Cosimo Laneve. Formal molecular biology. *Theoretical Computer Science*, 325(1):69–110, 2004.

[10] S. Geman and M. Johnson. Probabilistic grammars and their applications. *International Encyclopedia of the Social & Behavioral Sciences*, pages 12075–12082, 2002.

[11] D. Gillespie. Exact stochastic simulation of coupled chemical reactions. *Journal of Physical Chemistry*, 81:2340–2361, 1977.

[12] H. Hansson and B. Jonsson. A logic for reasoning about time and probability. *Formal Aspects of Computing*, 6(5):512–535, 1994.

[13] D. Harel. A grand challenge: Full reactive modeling of a multi-cellular animal. *Bulletin of the EATCS , European Association for Theoretical Computer Science*, 81:226–235, 2003.

[14] N. Kam, D. Harel, H. Kugler, R. Marelly, A. Pnueli, E.J.A. Hubbard, and M.J. Stern. Formal modeling of c. elegans development: A scenario-based approach. In *Comp. Methods in Systems Biology (CMSB)*, volume 2602 of *LNCS*, pages 4–20, 2003.

[15] M. Kwiatkowska. Model checking for probability and time: From theory to practice. In *Proc. of LICS'03*, pages 351–360. IEEE CS Press, 2003.

[16] 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.

[17] H. Matsuno, A. Doi, M. Nagasaki, and S. Miyano. Hybrid petri net representation of gene regulatory networ. In *Pacific Symposium on Biocomputing*, volume 5, pages 341–352, 2000.

[18] M. Nagasaki, S. Miyano, S. Onami, and H. Kitano. Bio-calculus: Its concept and molecular interaction. *Genome Informatics*, 10:133–143, 1999.

[19] L. Popova-Zeugmann, M. Heiner, and I. Koch. Modelling and analysis of biochemical networks with time petri nets. In *13th Int. Workshop on Concurrency Specification and Programming (CS&P'04)*, number 170 in Informatik–Berichte, pages 136–143. Humboldt–Universitaet, 2004.

[20] C. Priami, A. Regev, E.Y. Shapiro, and W. Silverman. Application of a stochastic name-passing calculus to representation and simulation of molecular processes. *Information Processing Letters*, 80:25–31, 2001.

[21] PRISM Model Checker. web site, 2004. http://www.cs.bham.ac.uk/˜dxp/prism.

[22] A. Regev. *Computational Systems Biology: a Calculus for Biomolecular Knowledge*. PhD thesis, Tel Aviv University, 2002.

[23] A. Regev, W. Silverman, and E.Y. Shapiro. Representation and simulation of biochemical processes using the pi-calculus process algebra. In *Pacific Symposium on Biocomputing*, pages 459–470, 2001.

[24] SICStus prolog. web site, 2004. http://www.sics.se/sicstus/.