Experiments with Reaction Systems

Allison Thompson Brown
University of Colorado at Boulder
430 UCB
Boulder, CO, 80309
allison.thompson@colorado.edu

ABSTRACT

Reaction systems provide functional models of real-world interactions between biochemical reactions. The parameters of these models can be tweaked to create systems that are predicted to rarely die. This paper gives an overview of the reaction system model and discusses the software we are building to experimentally verify theoretical predications and to answer additional questions about reaction systems. It also outlines a few new variations for simple reaction systems that we are examining.

Keywords

natural computing, biochemical reactions, reaction systems, random functions

1. INTRODUCTION

Reaction systems are important because they provide a formalized model of the biochemical reactions that are at the heart of functionality for a living cell, a model based on assumptions unlike those made by traditional computational models. Reaction systems are based on the observation that the two basic mechanisms behind the functions of biochemical reactions are facilitation and inhibition. They are also large, discrete systems and thus, to study them, experiments and computational simulations are essential for expanding our knowledge. In the course of our experimentation, we have encountered some results that are unexpected, and unpredictable. With a few small specifications, a simple reaction system "lifespan" is consistent in length with that of a totally random function over a domain roughly 75% the size of the reaction system's domain – but reaction systems *aren't* totally random. We don't know why this is, or what the implications are, and that is precisely the reason further experiments are necessary.

This paper gives an overview of the previous work with reaction systems by Eherenfeucht, Rozenberg, and Main. It continues by explaining the basic reaction system model, as well as discussing other previous work in more detail. It follows by discussing the software we are building to experimentally verify theoretical predications and to answer additional questions about reaction systems. It will also outline some future experiments we have planned with new variations on simple reaction systems, which we hope will shed new light on the reason(s) for the peculiar lifespan of simple reaction systems.

2. RELATED WORK

Authors Ehrenfeucht and Rozenberg introduce the concept of and notations for simple reaction systems in [4]. The formal model that reaction systems provide for biochemical reactions and their interactions with each other is based on a number of assumptions and axioms that are very different -- even orthogonal -- to those underlying traditional computational models. The model assumes "reactions are primary, while structures are secondary" [4]. It also assumes that "there is a 'threshold supply' of elements: either an element is present and then there is 'enough' of it, or an element is not present", and therefore there is no counting in the model. Furthermore, they assume there is no 'permanency" of elements: if the element is not produced by a reaction within a particular cycle, then it does not appear in the following cycle. Thus, sustaining an element's presence requires an effort [4].

In [2], Ehrenfeucht and Rozenberg introduce the concepts of events and modules to reaction systems, and investigate formation and evolution of those modules. The authors also prove that reaction systems can be viewed as self-organizing, where the organizing goal is to ensure a specific property of the set of all modules (the state of a process) [3]. They also present the idea of creating an "extended reaction systems" by adding a restriction relation to reaction system. The restriction relation determines which pairs of sets can form consecutive states in state sequence. They also note that, unlike in simple reaction systems, the idea of element permanency is intrinsic for an extend reaction system.

Ehrenfeucht and Rozenberg introduce the concept of time in reaction systems in [3]. Specifically, within the updated framework the notions of reaction times, compound creation, system life span, and other similar concepts can be created and defined.

In [1], Main, Ehrenfeucht, and Rozenberg present several formulae and theorems for use in predicting trajectory length for individual reaction systems in several different cases.

3. SIMPLE REACTION SYSTEMS

A reaction system is a functional model of the interaction between biochemical reactions. A simple reaction system consists of a number of reactions that interact with elements in the system's environment. It is formally defined as an ordered pair $\mathcal{A} = (S, A)$, "such that S is a finite set [of elements], and A is a finite set of reactions in S" [1, 2]. Individual reactions are made up of reactants, inhibitors, and products. Each of a reaction's reactants must be present for the products to be produced, but the presence of *any* of the inhibitors prevents the reaction from taking place. Furthermore, there can be no intersection between the reactants and the inhibitors; no element can be in both sets.

Formally, a reaction is defined as "an ordered triplet of nonempty, finite sets $a = (R_a, I_a, P_a)$, such that $R_a \cap I_a = \emptyset$." The elements that make up the reaction's components (reactants, inhibitors, and products) must be drawn from the system's environment, or background set (called S).

The list of elements present in the environment at any given point in a reaction system's life is the system's state, which we denote

Table 1. Average trajectory sizes for systems varying β.

β	n	Number Lived (out of 100,000)	Average trajectory size for those that lived	Number died (out of 100,000)	Average trajectory size for those that died
2.0	20	79,562	16.7	20,438	10.7
4.0	20	99,494	132.9	546	83.4
5.5	20	99,967	257.5	33	200.8
5.5	24	99,994	681.6	6	555.7
5.5	28	100,000	1,836.4	0	—
5.5	32	100,000	5007.4	0	—
5.5	36	100,000	13,575.5	0	—
5.5	40	100,000	36,974.7	0	—
5.5	44	100,000	101,643.9	0	—
8.0	20	98,576	179.9	1,424	124.1
16.0	20	45,394	25.1	54,606	16.6
32.0	20	2,436	5.2	97,564	3.9

as M. Each such state is a subset of the background set S. In our experiments, a reaction system begins "life" with a randomly generated M unless otherwise specified. The set of reactions for a system is also randomly generated, though each experiment has some control over this process. There may be one or more types of reaction within a reaction system; types of reactions are described in the following paragraph.

An (r,i)-reaction is a reaction with r reactants, i inhibitors, and a single product. So, a (3,1)-reaction is one with 3 reactants and 1 inhibitor. Reactions with multiple products can be normalized to (r,i)-reactions as follows: let $\{r_1, r_2 \ldots r_R\}$ be the set of reactants, $\{i_1, i_2, ..., i_I\}$ be the set of inhibitors, and $\{p_1, p_2, ..., p_P\}$ the set of products. Then P new reactions can be created, each of which has all R reactants and I inhibitors described, but only one element from the set of P products. Thus, the normal form for reactions is this (r,i) form, always with a single product.

If all of a reaction's reactants are present in the reaction system's state, and each inhibitor is absent, that reaction is *enabled* by the state. Formally, for a reaction a, if $I_a \cap M = \varnothing$ and $R_a \subseteq M$, then we say that *a is enabled by M*. Each iteration of a reaction system's life follows the same pattern: if a reaction in the system is enabled, its products are added to the next state. Elements not produced by the system's current enabled reactions are dropped from the state, and do not carry over. There is also no concept of an amount of an element – in each state, each element is either present or it is not. If an element is not produced by some enabled reaction, it will not be added to the next state.

This process is formally defined as follows: the result of a reaction a on the next state of the system is a function $res_a = 2^S \rightarrow 2^S$, such that 2^S is the power-set of elements in S. So, $res_a(M) = P_a$ if a is enabled by M, and the empty set otherwise. Furthermore, the system's next state is equal to the union of $res_a(M)$ for all reactions $a \in A$.[1] The state that follows M can be denoted as $\mathfrak{A}(M)$, also called "the effect of running the system $\mathfrak{A}$ one iteration on M."

4. EXPERIMENTS WITH LIFE AND DEATH

A system's *trajectory* is a list of all states that system has visited, including the start state. If the state is already in the trajectory, it is not added a second time. In this case, the system has entered a loop. A system that exhibits this behavior is one that "lives" forever, since there is no event that can cause the system to exit its loop. Alternatively, if an empty state is reached, that system is said to be "dead," since there is now no way for it to have any enabled reactions. A trajectory's length is the number of states that trajectory contains.

A reaction system constructed completely at random tends to die rather than enter a trajectory loop. By using parameters based on the approximations described in [1], it is possible to create systems that are predicted to rarely die. Specifically, theoretical results [1] predict that systems with n elements, $5.5n$ (1,3)-reactions, and $5.5n$ (3,1)-reactions will quickly move to a state with about $n/2$ elements and stay there, never dying. In fact, in experiments for systems with these parameters, in all cases where $|S| \geq 28$, the system does not die.

The full results of experimenting with varying numbers of reactions can be found in Table 1. The number of reactions is controlled by a variable β, such that a system has βn of each reaction type. Results are shown for a variety of β values with fixed $n = 20$, with detailed results for the optimal value $\beta = 5.5$.

5. TRAJECTORY LENGTH PREDICTION

In the course of experimentation, it was also discovered that reaction system trajectory sizes could be predicted using Donald Knuth's Q-function for random functions [5][6]. If random function f is defined such that f: $D \rightarrow D$, for a finite domain D of size N, then its trajectory set size can be predicted by the Q-function, such that

$$Q(N) = \sqrt{\frac{\pi}{2}}N^{0.5} - \frac{1}{3} + \frac{1}{12}\sqrt{\frac{\pi}{2}}N^{-0.5} - \frac{4}{135}N^{-1} + \frac{1}{288}\sqrt{\frac{\pi}{2}}N^{-1.5} + O(N^{-2})$$

Furthermore, Knuth shows that $Q(N)$ can be closely approximated by the first term in the sum, $\sqrt{\pi/2}N^{0.5}$ [5].

If reaction systems behaved entirely like totally random functions (which they are not), then their trajectory sizes could be predicted by calculating $Q(2^n)$. From our experiments, reaction system trajectories tend to be shorter than that prediction, but *can* be predicted by calculating $Q(2^{0.74n})$, as is shown in figure 1. It appears that reaction systems share some behavior with completely random functions on domains of approximately $2^{0.74n}$ elements. Our first experiment to determine the explanation for this apparent similarity is detailed in section 5.1.

[1] Where A is the set of reactions in a reaction system, $\mathfrak{A} = (S, A)$.

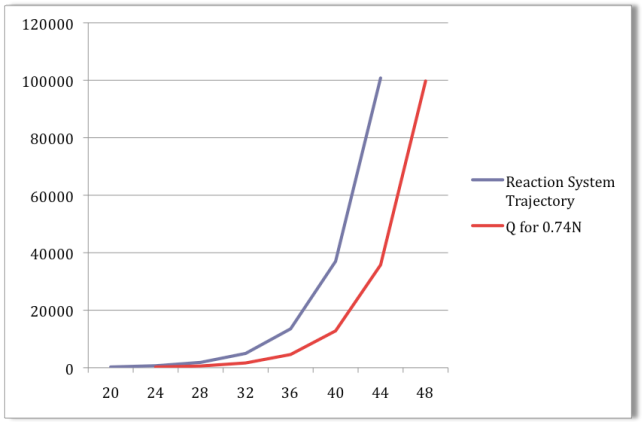

Figure 1. A comparison of actual reaction system trajectory size for N=20 to N=48 to predictions made by Q for 0.74N.

5.1 Potential Explanation: Correlation between Background Set Elements

Our current research is investigating why the unexpected relationship between reaction systems and totally random functions holds. Initially, we thought that it might be the case that the restrictions we place on the systems we generate cause a correlation between the elements of S such that the actual size of the background set was closer to $0.74n$. That is to say, if two elements had similar or identical reactants and inhibitors, then the correlation of those two elements would be higher than chance. If the correlation between them were perfect, the size of the S would be effectively reduced to n-1. We were curious to discover if the correlations between elements was high enough to reduce the number of background set elements to $0.74n$.

We tested our hypothesis by computing the eigenvalues for the matrix of correlation coefficients between the elements of S. If, for example, the matrix had i small eigenvalues and n-i large ones, that would indicate that the actual size of S was n-i. Unfortunately for our potential explanation, none of the systems we tested had fewer than n eigenvalues; in fact, each had exactly n, all of similar sizes. We could only conclude correlation between background set elements does not explain our findings.

As a result of this, much of our current work is centered around the exploring the Q-function quirk; specifically, we create new types of reaction systems in order to discover if they share this unexpected behavior, and thus to learn more about why we have encountered these results. We also motivated to create new types of systems because, in addition to discovering the reason for the apparent similarity between reaction systems and totally random functions, we would like to be able to predict trajectory lengths in a similar fashion for randomly-generated reaction systems with differing numbers of and varying types of reactions.

6. SOFTWARE FOR VARIATIONS ON SIMPLE REACTION SYSTEMS

The software we've written to run the experiments consists of a hierarchy of C++ classes that allows for easy experimentation with reaction systems. The hierarchy is also designed to be easily expandable with new system types.

The base class for the hierarchy is the "Mapping" class, which encompasses all traits that the systems we use have in common, such as the current state. It is subclassed by the "UniformMapping" class and the "AbstractReactionSystem" class.

6.1 Uniformly-Mapped Systems: The UniformMapping Class

A system implemented by the UniformMapping class contains no reactions, and instead mimics the behavior of a completely random function, generating random states until it loops or creates an empty state and the system "dies." It is used solely for comparative purposes, and to verify the predictions made by Knuth's Q-function about the length of a totally random function's trajectory.

6.2 Reaction System Classes

The AbstractReactionSystem class is, as its name implies, an abstract class not intended for implementation. Instead, it provides a common library that the classes that inherit from it can use. The PseudoReactionSystem class and the ReactionSystem class both inherit from the AbstractReactionSystem class.

6.2.1 Non-uniformly Mapped Systems: The PseudoReactionSystem Class

The PseudoReactionSystem class provides an alternate way to implement non-uniformly mapped systems. While it might seem closer to the uniformly mapped systems described previously, it is in fact a closer cousin to simple reaction systems. These systems create probability tables for reaction system state sizes based on the approximations from [1]. The initial state is randomly generated, and subsequent state sizes are determined via look-up in the table. So, if the current state size is i, and there is a 40% chance, according to the table, that the next size will be j, then there is 40% chance that the next state size will be j.

6.2.1.1 Trajectory Prediction

Pseudo-reaction systems tend to have much longer trajectories than their simple cousins. Experimental results have shown their trajectories to be as much as 15.1 times longer, a median of 8.67 times longer, and an average of 9.23 times longer. The Q-function, which was some help in predicting the trajectory size for simple reaction systems, is less useful for pseudo-reaction systems. Though it initially appeared that pseudo-reaction systems had trajectory sizes that converged on $Q(2^{0.93n})$ as n approached infinity,[2] that initial trend was disproved. After an initial plateau, trajectory sizes continued to shrink when compared with the Q-function. The smallest recorded size is currently approximated by $Q(2^{0.71n})$, for n = 56. At that point, our results appear to indicate that the system no longer enters a cycle. Further experimentation will be necessary to determine if this is the case, and why such behavior is happening, in addition to why Q drops below the 0.74n limit of the simple reaction systems.

6.2.2 Simple Reaction Systems: The ReactionSystem Class

The ReactionSystemClass provides a way to implement a reaction system as described in Section 2. Some of our current work involves creating a new class to inherit from this one. We are currently examining the effects of adding background element

[2] Many thanks to the anonymous reviewer whose comments helped me discover that my original calculations here were incorrect.

persistence to simple reaction systems in a new form of system we call "reaction systems with duration." These new systems are identical in construction to simple reaction systems except for a variable *H*, which determines how long each element of *S* remains in the system environment after it has first been created. While it looks as though this new system type will provide interesting new data, our initial results have not yet been verified, and so no substantial conclusions can be drawn.

7. FUTURE WORK

Our next steps will be to verify the initial results gathered from experimentation with reaction systems with duration. Should our results prove valid, we will perform further experiments to fully explore the properties of the new system type. We will also continue to investigate the theoretical basis for the results described in Section 5.

8. ACKNOWLEDGMENTS

I would like to thank Dr. Andrzej Ehrenfeucht and Dr. Michael Main for their advice and guidance of the work here described. I would also like to thank the anonymous reviewers for their numerous suggestions and comments.

9. REFERENCES

[1] Ehrenfeucht, A., Main, M., and Rozenberg, G. 2010. Combinatorics of Reaction Systems. *International Journal of Foundations in Computer Science*, *21*, 3 (June 2010), 345-356.

[2] Ehrenfeucht, A. and Rozenberg, G. Events and Modules in Reaction Systems. *Theoretical Computer Science*. *376*, 1-2 (2007), 3-16.

[3] Ehrenfeucht, A., and Rozenberg, G. Introducing Time in Reaction Systems. *Theoretical Computer Science, 410*, (2009), 310-322.

[4] Ehrenfeucht, A. and Rozenberg, G. Reaction Systems. *Fundam. Inf.* 75, 1-4 (January 2007), 263-280.

[5] Knuth, D.E. *The Art of Computer Programming v.1: Fundamental Algorithms*. 116-121. Addison-Wesley, Reading, MA, 1968.

[6] Knuth, D.E. *The Art of Computer Programming v. 2: Seminumerical Algorithms*. 8, 454-5.Addison-Wesley, Reading, MA, 1969.