in: Second German Workshop on Artificial Life
Draft Proceedings
pp. 19-22, University of Dortmund, 1997

Macroscopic and Microscopic Computation in an Artificial Chemistry

Peter Dittrich, Wolfgang Banzhaf, Hilmar Rauhe and Jens Ziegler
University of Dortmund, Dept. of Computer Science, D-44221 Dortmund (Germany)
http://ls11-www.informatik.uni-dortmund.de/

1 Introduction

Chemical and biochemical systems as part of living organisms have been shown to possess interesting computational properties [1, 2, 7]. As an example Figure 1 shows the information flow in the chemotaxis system of Escherichia Coli [10]. In a parallel development, the chemical computation metaphor is becoming more and more frequently used as part of the emergent computation paradigm in Computer Science [3, 4, 5, 6, 8]. In this contribution we will discuss two ways of how information can be processed by a collection of molecules floating around in well-stirred tank reactor. In the first case the information is stored as a concentration of a substances and computation is carried out by increase and decrease of concentration levels. We will refer to this as **macroscopic computation**.

In the second case – **microscopic computation** – the result of a computation is represented by single molecules. The dynamics is stochastic, in contrast to macroscopic computation where the dynamics can be described with ordinary differential equations [11].

In both cases the result emerges from many simple and parallel interactions. In order to show the abilities of such systems we will use **artificial chemistries** which are simulated reaction systems of mathematical or algorithmic objects.

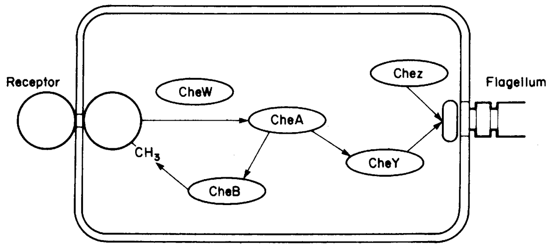

Figure 1: Information flow in the chemotaxis system of *Escherichia Coli*. Binding of stimulus molecules causes activation of CheA by CheW. CheA activates CheB which resets the receptor thereby closing the feedback loop. CheY binds to the switch complex and causes clockwise rotation. It is antagonized by CheZ. Diagram from [9].

2 Example for Microscopic Chemical Computation: Prime Number Generator

In this example we set up a reaction system which produces prime numbers. In addition we will demonstrate that the population size is critical for a successful computation. When increasing the population size a phase transition occurs with respect to the qualitative behavior of the system, i.e. to the production of prime numbers.

Starting with a pool of M integer numbers selected randomly from the set $\{2, 3, \ldots, max\}$ with $M << max$ the following must be considered:

1. Prime numbers should be generated either directly or through intermediate numbers.

2. Prime numbers and intermediate numbers should have advantages over numbers with many prime factors.

The first point is taken into account by using the mathematical division as the collision or reaction rule. More precisely, the product s_3 of the collision $s_1 + s_2 \Longrightarrow s_3$ is defined as

$$s_3 = s_1/s_2 \qquad \text{if } s_1 \bmod s_2 = 0 \text{ and } s_1 \neq s_2 \ .$$

Otherwise the collision is "elastic" and no result occurs. A number s_1 is only replaced if a reaction $s_1 + s_2 \Longrightarrow s_3$ successfully produces a number s_3.

Figure 2 shows a simulation where the population size M is set to 100 and $max = 10000$. The concentration of prime numbers increases until no non-prime is left.

Figure 3 demonstrates that the soup size could be critical. As always, it is decisive to have reaction pathways at hand which do not lead into dead ends but keep enough reaction paths open toward the state aimed at (high prime number concentration). Here the numbers can be viewed as simple machines that manipulate other numbers by dividing them. Too small a soup size would not provide enough useful "machines" (e.g. numbers that are able to divide other numbers) at the beginning to initiate the avalanche-like process depicted in Figure 2.

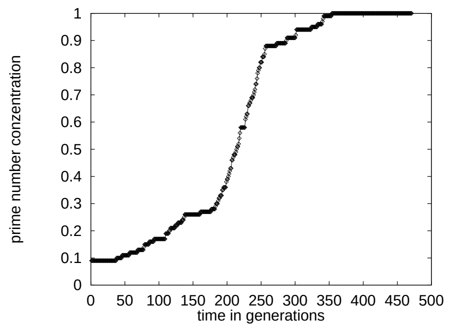

Figure 2: A single run for the prime number problem where at last every object inside the reactor is a prime number. The soup is initialized with random numbers out of $\{2, \ldots, 10000\}$. Soup size: $M = 100$.

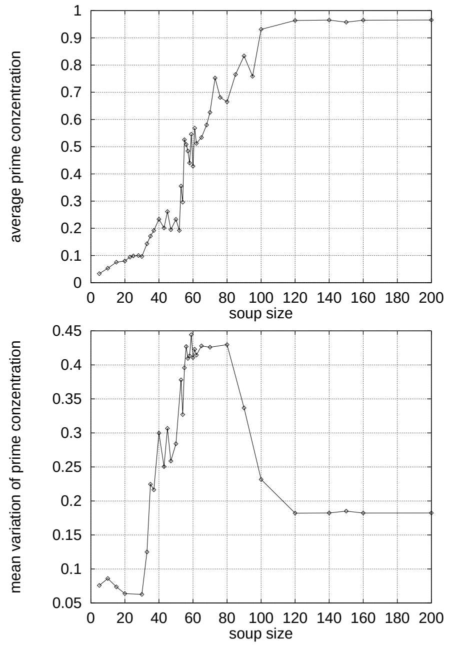

Figure 3: Dependence of prime number production on the population size. The figures summarize a series of runs with different soup sizes. Each dot represents 30 runs with the same soup size M. In the first graph the average prime concentration at the end of each run (generation 700) is displayed. For small soup sizes ($M < 40$) the algorithm runs mostly into a dead end so that the soup still contains a lot of non-prime numbers. With a large soup size ($M > 100$) the reactor is nearly always able to transform every number to prime number resulting in a prime number concentration of 100%. In the vicinity of the phase transition ($40 < M < 100$) a strong fluctuation of the number of primes each run produced can be observed. This results in the second figure where the mean variation of a single run is depicted.

3 Example for Macroscopic Chemical Computation: Robot Control

In this example the principle of macroscopic chemical computation should be demonstrated by applying it to the problem of controlling an autonomous robot. Figure 4 shows the relationship of the robot and its environment. Here, the flow of data would correspond to the flow of substances and processing of the data, i.e. computation, would correspond to chemical reactions which are carried out in a simulated well-stirred tank reactor inside the robot.

Figure 5 shows a simple hand-written reaction network. In consists of 15 substances. Sensory information is inserted into the system as "sensor substances" (e.g. AW, AN etc.). These are processed by "working substances" leading to concentration changes in the "motor substances". The motor system is able to measure the concentration of the "motor substances" and moves the robot accordingly.

Current work aims at building a robot controller based on a more general artificial chemistry consisting of a potentially infinite number of different substances. In addition, a topology will be introduced where different compartments are connected via diffusion. Figure 6 and 7 show preliminary results.

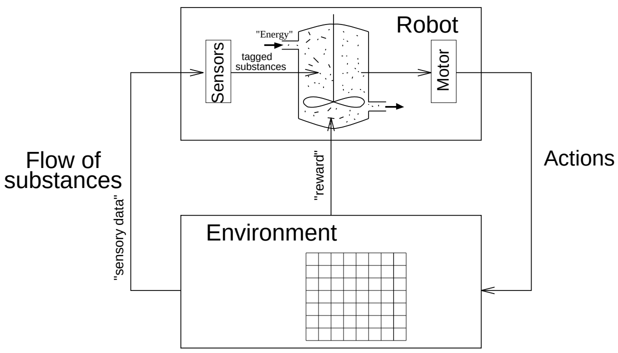

Figure 4: The robot and its environment. Here the transmission of sensory data is interpreted as a flow of substances. The processing of these substances is achieved by an algorithmic well stirred tank reactor.

4 Discussion

In this contribution it is demonstrated on an abstract level how computation can be achieved by chemical systems. Despite the fact that the reactors presented in this paper make use of artificial interactions they point to a strategy for the implementation of parallel algorithms in molecular computation and give insight into the reactor's design and system dynamics that will determine the behavior of in vitro reactors.

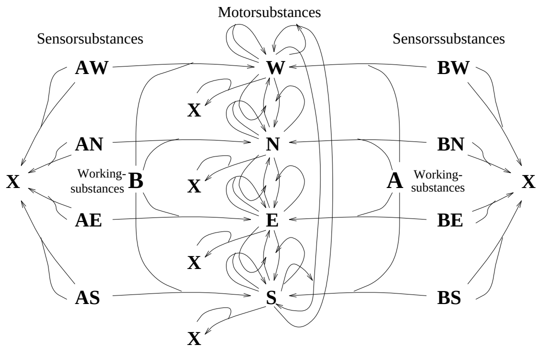

Figure 5: Hand-written metabolic network of a robot controller.

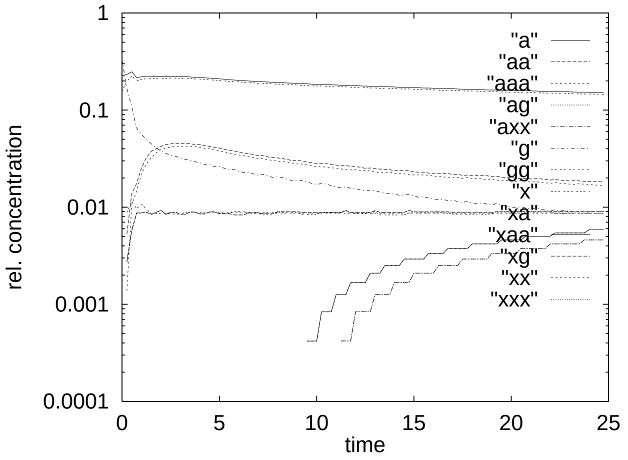

Figure 6: Development of substance concentration of one compartment. Polymerisation builds new species and the system finally reaches steady state. Initial population is $\{a, g, x\}$. Final population is $\{a, aa, aaa, ag, g, gg, x, xa, xaa, xg, xx, xxx\}$. The catalytic efficiency is computed by a string matching algorithm. Species can only catalyse the polymerisation of other species.

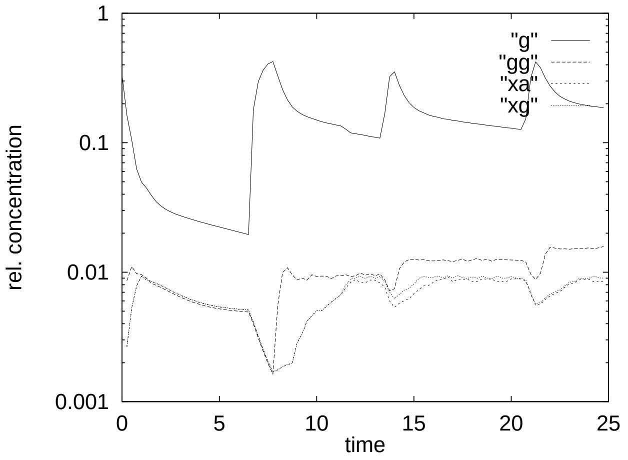

Figure 7: The concentration of g exceeds the threshold and it participates in the reaction system. Immediately after the increase of g the polymerisation product gg increases, particularly at $t \approx 7.5$ and $t \approx 14$.

Acknowledgments

This project is supported by the DFG (Deutsche Forschungsgemeinschaft) under grant Ba 1042/2-1.

References

[1] L.M. Adleman, *Molecular Computation of Solutions to Combinatorial Problems*, Science, **266** (1994) 1021

[2] A. Arkin and J. Ross, *Computational Functions in Biochemical Reaction Networks*, Biophys. J., **67** (1994) 560

[3] W. Banzhaf. Self-organizing algorithms derived from rna interactions. In Wolfgang Banzhaf and Frank H. Eeckman, editors, *Evolution and Biocomputing*, Volume 899 of *Lecture Notes in Computer Science*, pages 69–103. Springer, Berlin, 1995.

[4] W. Banzhaf, P. Dittrich and H. Rauhe. Emergent computation by catalytic reactions. *Nanotechnology*, 7: 307–314, 1996.

[5] G. Berry, G. Boudol, *The chemical abstract machine*, Theor. Comp. Sci. **96** (1992) 217

[6] W. Fontana. Algorithmic chemistry. In Christopher G. Langton et al. ed., *Artificial Life II proceedings*, pages 159–209. Addison-Wesley, Redwood City, CA, 1992.

[7] A. Hjelmfelt, E.D. Weinberger and J. Ross, *Chemical implementation of neural networks and Turing machines*, Proc. Natl. Acad. Sci. USA, **88** (1991) 10983

[8] T. Ikegami and T. Hashimoto. Coevolution of machines and tapes. In F. Morán, et al. ed. *Artificial Life III proceedings*, volume 929 of *LNAI*, pages 234–245. Springer Verlag, Berlin, 1995.

[9] Jones, C. and S. Aizawa (1991). The Bacterial Flagellum and Flagellar Motor: Structure, Assembly and Function. *Advances in microbial physiology 32*, 108 –171.

[10] Lengeler, J. W. (1996). Die biologische Zelle als komplexes Informationssystem. *Biologen in unserer Zeit 426*, 65 – 70.

[11] P. F. Stadler, W. Fontana, and J. H. Miller. Random catalytic reaction networks. *Physica D*, 63:378–392, November 1993.