Molecular computers in artificial chemistry

Marius Buliga

Institute of Mathematics, Romanian Academy
Marius.Buliga@imar.ro, mbuliga@protonmail.ch

AMA (ask me anything) about artificial chemistry experiments with chemlambda, lambda calculus or interaction combinators, described in arXiv:2003.14332, available from the page chemlambda.github.io.

(My secret hope is that you also ask me about the mathematical source of this preoccupation, emergent algebras or dilation structures on metric spaces [1] [2].)

Here we think about an artificial chemistry as a triple (Graphs,Rewrites,Enzymes) where
- Graphs is an infinite class of molecules, a molecule is a port graph in the sense of Bawden with nodes decorated with a finite set of node types
- Rewrites is a class of DPO graph rewrites on molecules of Graph
- Enzymes is a Turing machine which can perform the graph rewrites on molecules locally (by using a finite work tape) and randomly.

The interest is to explore if such chemistries can compute, according to one of the following meanings:

1. given a parser from lambda calculus [or your language of preference] to molecules as an input device and a decorator of molecules edges with lambda terms [or your laguage of preference] as an output device, can the artificial chemistry produce the expected output [all the time? most of the time? with a probability greater than some constant?]

2. given as input a (class of molecules) and as goal a class of global rewrites (like global fan-out, i.e. duplication, or global isomorphism, i.e. metabolism, or productions of another class of molecules), can we use a lambda calculus [or your language of preference] inspired artificial chemistry to achieve the goal [all the time? most of the time? with a probability greater than some constant?]

Details. Molecules are encoded as mol patterns, which are vectors of mol nodes, where each mol node is a vector of node types and tags of edges connected to the node ports.

The enzyme machine has two tapes: the IO tape and the work tape. The tapes of the machine have cells, each cell of a tape can contain a node type, a port type, an edge tag or a finite set of special characters.

The machine can read from on the IO tape a cell, write a whole mol node at the end of the mol pattern, or delete a whole mol node. For each read/write or delete from/to the IO tape the machine writes the same on the work tape. The machine can read from the work tape and it can write only on the blank cells of the work tape. The machine can either halt (by writing a halt symbol on the work tape) or it can delete the whole work tape and then the IO head jumps to a random mol node beginning.

The constraints are:

- the machine can perform any graph rewrite from Rewrites on a molecule from Graphs
- at each moment the IO tape contains only a well written mol pattern
- the work tape has finite length.