

A new molecular programming language: CRN++

5 October 2018, by Ingrid Fadelli

Euclid's Algorithm:

```

1: procedure GCD(a,b)
2:   while a ≠ b do
3:     if a > b then
4:       a ← a - b
5:     else
6:       b ← b - a
7:     end if
8:   end while
9:   return a
10: end procedure

```

CRN++ Program:

```

GCD = {
  conc[a,a0],
  conc[b,b0],
  step[{
    Ld[a, atmp],
    Ld[b, btmp],
    Cmp[a,b]
  }],
  step[{
    IfGT[{{ Sub[atmp,btmp,a] }},
    IfLT[{{ Sub[btmp,atmp,b] }},
  ]}
];

```

The Euclid's algorithm and how it will be written in CRN++. Credit: Vasic et al.

Synthetic biology is a relatively new area of research that could significantly impact a number of fields, including biology, nanofabrication and medicine. A primary challenge in this emerging field is that of embedding computation in molecular contexts, in situations where electronic micro-controllers cannot be inserted. Doing this requires the development of methods that can effectively represent computations using molecular components.

A team of researchers at The University of Texas at Austin has created CRN++, a new [language](#) for programming deterministic (mass-action) [chemical kinetics](#) in performing computations. In their paper, pre-published on *arXiv*, the researchers outline this new language and build a compiler that translates CRN++ programs into [chemical reactions](#).

"A key technical challenge of [synthetic biology](#) is to design a chemical controller that interacts within a cellular environment, achieving a particular task," Marko Vasic, one of the researchers who carried out the study, told *Tech Xplore*. "To achieve this, it is necessary to both engineer synthetic molecules

and [program](#) them. Molecules interact via chemical reactions, and by programming molecules, we mean defining interaction rules (chemical reactions) between them."

Recent improvements in DNA synthesis have opened up new, exciting possibilities for molecule engineering. However, researchers in synthetic biology first need to devise ways to design the interaction rules (chemical reactions) to achieve a desired goal. The main objective of this recent study was to design a higher-level language that could express the behaviour of chemical reactions in a more intuitive way.

"In software engineering, a programmer writes in a high-level language that is easy to understand, and such a program is compiled down to the machine code, which is hard to understand by a human, but understandable by a machine," Vasic explained. "We tried to build an analogy in molecular programming by defining a high-level language that is easier to reason about and is compiled down to 'complex' chemistry."

Compiler Output:

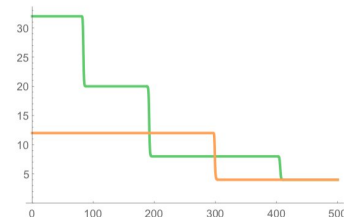
```

A + X[3] -> A + Atmp
Atmp + X[3] -> T + X[3]
...
B + XGTy + X[3] -> B + XLTY + X[3]
O + XLTY + X[3] -> O + XGTy + X[3]
O + YLTx + X[3] -> O + YGTx + X[3]
...

```

CRN

CRN Simulation:



GCD for a=32 (green), b=12 (orange)

The output of the compiler as well as simulation results for the CRN++ program. Credit: Vasic et al.

CRN++ is based on two ideas: modularity and the use of an oscillator. Modularity means that the

language includes a set of chemical reactions called modules that can be composed with no interference between different sets of reactions. To achieve this, the researchers mapped basic operations of CRN++ to these modules. They also used a chemical oscillator for temporal ordering, which allowed them to translate ordered imperative commands of the language into chemistry.

"To the best of our knowledge, we are the first to provide an imperative programming language that complies with chemical [reaction](#) networks," Vasic said. "We open-sourced our code, including CRN++, as well as simulation framework, as we hope that this will make it easier for researchers to try out new approaches, and thus advance the area further."

The researchers evaluated CRN++ and proved its feasibility on a series of algorithms for discrete and real-value computation. The new language can also be easily extended to support new commands or implementations, making it the ideal basis for the development of more advanced molecular programs.

"Programs translated from CRN++ to chemistry contain some amount of error, which can be very low in some classes of programs but can be high or build up over time in other ones," Vasic said. "We hence plan to investigate sources of error further and design programs that guarantee that error does not build up over certain bound."

Vasic and his colleagues are also looking to broaden their programming language by including new modules, defined as sets of chemical reactions that can carry out basic operations.

More information: CRN++: Molecular Programming Language. arXiv:1809.07430 [cs.ET]. arxiv.org/abs/1809.07430

© 2018 Tech Xplore

APA citation: A new molecular programming language: CRN++ (2018, October 5) retrieved 14 May 2021 from <https://techxplore.com/news/2018-10-molecular-language-crn.html>

This document is subject to copyright. Apart from any fair dealing for the purpose of private study or research, no

part may be reproduced without the written permission. The content is provided for information purposes only.