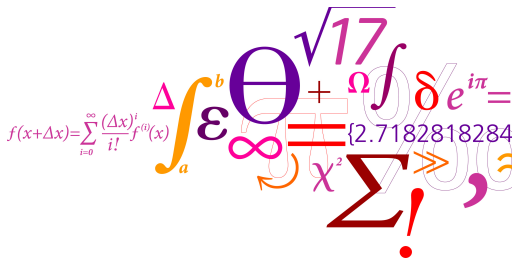


02257 Applied Functional Programming

Brief introduction to Project 2

Michael R. Hansen



DTU Compute

Department of Applied Mathematics and Computer Science

Teaching activities: tentative plan from June 14-26

Friday, June 14

- 9:15 - Approx 11:00:
Intro Project 2
Intro to Parsing using combinators

Monday, June 17

- 9:15 - 11:15: Roar is available in Room 041B

Tuesday, June 18:

- 11:15 - 12:00: Lecture on Property-based testing

Wednesday to Friday, June 19 - 21:

- 9:15 - 11:15: Roar is available in Room 041B

Monday to Wednesday, June 24 - 26:

- 9:15 - 11:15: Roar is available in Room 041B

Based on the paper:

- M. Vasic, D. Doloveochick and S Khurshid: CRN++: Molecular programming language
Natural Computing 19, pages 391-407, 2020

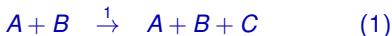
A high-level programming language designed for

- compilation to chemical reaction networks

Facilitates computations in contexts where microprocessors cannot be inserted.

Computation by chemical reactions: An example

A chemical reaction network for multiplication: $C := A * B$:



Two reactions:

- $A + B$ is the **reactant** of (1)
- $A + B + C$ is the **product** of (1)
- both with **rate** 1

With large molecular counts and well-mixed solutions we can model dynamics with ordinary differential equations:

$$d[A]/dt = 0$$

$$d[B]/dt = 0$$

$$d[C]/dt = [A](t) \cdot [B](t) - [C](t)$$

See Fig. 1 in the paper for visualization

A CRN++ example: GDD

```
crn =  
  { conc[a,32], conc[b,12];  
  
    step[{ ld[a,atemp], ld[b,btemp], cmp[a,b] }] ;  
  
    step[{ ifGT[{ sub(atemp,btemp,a) }] ,  
            ifLT[{ sub(btemp,atemp,b) }] ]  
  }
```

Fig 3(a,b)

The program consists of

- ① Initialization of concentration of species
- ② Two steps.
 - ① The commands in a step must be non-conflicting and are executed in parallel
 - ② The steps are executed in sequence

The two steps are repeated indefinitely

- Visualization of the dynamics

Compilation to chemical reaction networks

The article describes principles and techniques for compiling CRN++ programs to chemical reaction networks.

Mix species and observe dynamics
everything happens in parallel

How can the reactions be controlled?

The paper introduces basic modules that

- use input catalytic
 - show exponential convergence
 - have a unique stable steady state
- combinations of such modules satisfy these properties as well.

The paper shows how sequencing can be obtained using chemical oscillators

See Fig 4.

Errors (non-exact computations) are unavoidable

The tasks of Project 2

Part 1: A “classical” interpreter for CRN++

- 1 Analyse the description of CRN++ in the article. Outcome is a revised grammar and a list of syntactic restrictions
- 2 Construct a parser and a function that checks whether a program is *well-formed*
- 3 Construct an interpreter and a visualization tool

Part 2: Chemical reaction networks

- 1 Parser
- 2 Simulator based on ODE's
- 3 Visualization tool

Part 3: Compilation from CRN++ to reaction networks in stages

- 1 Step
- 2 ...

Implement as much as possible from the article.

Validate your implementation using systematic testing, property-based testing and applications.