Calculs Analogiques dans les Programmes Biochimiques Naturels et Synthétiques

François Fages
Project-Team Lifeware
http://lifeware.inria.fr/

Institut National de Recherche en Informatique et Automatique
Inria Saclay – Ile de France
Cells Compute

They process signals

Regulate their metabolism

Take decisions such as
  – Replication
  – Differentiation
  – Migration

Understanding these processes is a central difficulty in many applications in medicine, health, agriculture and the ultimate goal of molecular cell biology.

François Fages

JLLL'50 Roscoff, Mars 2019
Chemical Reaction Networks (CRNs)

**CRN structure**: network of reactants, products, and reactions
- \(2H_2 + O_2 \rightarrow 2H_2O\) but not necessarily mass balanced
- synthesis reactions \(\_ \rightarrow A\) degradation reactions \(A \rightarrow \_\)
- hypergraph of reactions (bipartite species-reaction graph)

**CRN dynamics**: several interpretations
- differential equations, continuous-time Markov chains,
- Petri net, Boolean transitions

**CRN model repositories (Systems Biology Markup Language SBML)**:
- BioModels.net \(\approx\) 2000 models + 10000 models of metabolism

**CRN theory**: interplay between CRN structure and CRN dynamics
- static analysis of steady states, stable states, oscillations,…
- reductions by quasi-steady state (QSS) and quasi-equilibrium (QE) approximations
- Turing completeness and computational complexity of CRNs
Several Interpretations $A + B \xrightarrow{k. A. B} C$

**Continuous semantics:** concentrations, continuous time evolution

Ordinary differential equations (ODE)

$$\frac{dA}{dt} = -k \cdot A \cdot B \quad \frac{dB}{dt} = -k \cdot A \cdot B \quad \frac{dC}{dt} = k \cdot A \cdot B$$
Several Interpretations $A + B \xrightarrow{k.A.B} C$

Continuous semantics: concentrations, continuous time evolution

Ordinary differential equations (ODE)

\[
\frac{dA}{dt} = -k \cdot A \cdot B \quad \frac{dB}{dt} = -k \cdot A \cdot B \quad \frac{dC}{dt} = k \cdot A \cdot B
\]

Stochastic semantics: numbers of molecules, probability and time of transition

Continuous Time Markov Chain (CTMC)

$A, B \xrightarrow{p(S_i), t(S_i)} C++, A--, B--$

Intrinsic variability (with same genetic and epigenetic parameters)
Several Interpretations $A + B \xrightarrow{k.A.B} C$

Continuous semantics: concentrations, continuous time evolution

Ordinary differential equations (ODE)

$$\frac{dA}{dt} = -k \cdot A \cdot B \quad \frac{dB}{dt} = -k \cdot A \cdot B \quad \frac{dC}{dt} = k \cdot A \cdot B$$

Stochastic semantics: numbers of molecules, probability and time of transition

Continuous Time Markov Chain (CTMC)

$$A, B \xrightarrow{p(S_j), t(S_i)} C++, A--, B--$$

Multi-agent simulation: numbers of molecules, space, diffusion speed, affinity

Random walk (ex. Hsim simulator [Amar 04])
Several Interpretations $A + B \rightarrow^{k.A.B} C$

**Continuous semantics:** concentrations, continuous time evolution

Ordinary differential equations (ODE)

\[
\frac{dA}{dt} = -k \cdot A \cdot B \quad \frac{dB}{dt} = -k \cdot A \cdot B \quad \frac{dC}{dt} = k \cdot A \cdot B
\]

**Stochastic semantics:** numbers of molecules, probability and time of transition

Continuous Time Markov Chain (CTMC)

\[A, B \xrightarrow{p(S_i), t(S_i)} C++, A--, B--\]

**Petri net semantics:** numbers of molecules

Multiset rewriting

CHAM [Berry Boudol 90] [Banatre Le Metayer 86]
Several Interpretations $A + B \xrightarrow{k.A.B} C$

Continuous semantics: concentrations, continuous time evolution

Ordinary differential equations (ODE)

\[
\begin{align*}
\frac{dA}{dt} &= -k \cdot A \cdot B \\
\frac{dB}{dt} &= -k \cdot A \cdot B \\
\frac{dC}{dt} &= k \cdot A \cdot B
\end{align*}
\]

Stochastic semantics: numbers of molecules, probability and time of transition

Continuous Time Markov Chain (CTMC)

\[
A , B \xrightarrow{p(S_i), t(S_i)} C++, A--, B--
\]

Petri net semantics: numbers of molecules

Multiset rewriting

CHAM [Berry Boudol 90] [Banatre Le Metayer 86]

Boolean semantics: presence/absence

Asynchronous transition system

Symbolic model-checking

\[
\begin{align*}
A \land B &\rightarrow C \land \neg A \land \neg B \\
A \land B &\rightarrow C \land A \land \neg B \\
A \land B &\rightarrow C \land \neg A \land B \\
A \land B &\rightarrow C \land A \land B
\end{align*}
\]
Hierarchy of CRN Semantics

- **Boolean traces**
- **Discrete traces (Petri net)**
- **Continuous traces (ODE)**
- **Stochastic traces (CTMC)**

---

**Thm. (abstract interpretation)**
Galois connections between the syntactical, stochastic, Petri net and Boolean trace semantics
[Fages Soliman Theoretical Computer Science 2008]

**Thm. (approximation)**
For large numbers of molecules the ODE semantics approximates the *mean* stochastic behavior
[Gillespie 1971 Kurtz 1978]

---

If a behavior is not possible in the Boolean semantics (symbolic model-checking), it is not possible in the stochastic semantics for any reaction rates

Animal model

Synthetic microreactor

Model cell-to-cell variability
Intrinsic and extrinsic variability

JLLL'50 Roscoff, Mars 2019
Turing Completeness of CRNs?

**Discrete CRN:** Not Turing complete without
- Test of absence (Petri net inhibitor arc)
- Polymerisation reactions
  [Cardelli Zavatero MSCS 2010, Cook et al 2009]
- Unbounded membranes

**Stochastic CRN:** Simulation of a Turing machine with a small probability of error
  [Cook, Soloveichik, Winfree, Bruck 2009]

**Continuous CRN:** Non uniform computability: for each function for each input there exists a circuit computing the result

Strong Turing completeness?
MAPK Signalling Cascade

MAPK Signaling Network: **30 reactions 18 species** [Huang Ferrel PNAS 1996]
MAPK Input/Output Function

Dose-response diagrams alias Bifurcation diagrams

biocham: load(library:examples/mapk/mapk).

biocham: dose_response('E1',1.0e-6,1e-4,200).

MAPK responses as Hill function $\frac{x^n}{c + x^n}$

[Huang Ferrel 96 PNAS]

n $\approx$ 4.9 at 3$^{\text{rd}}$ level
n $\approx$ 1.7 at 2$^{\text{nd}}$ level
n = 1 at 1$^{\text{st}}$ level (Michaelis-Menten)

MAPK implements the function of an analog/digital converter in the cell.

How would one program $\frac{x^n}{c + x^n}$ with biochemical reactions?

What does it mean to compute with real numbers?
Computable Real Numbers and Functions

Classical definitions of computable analysis based on Turing machines

**Definition.** A real number $r$ is computable if there exists a Turing machine with
Input: precision $p \in \mathbb{N}$
Output: rational number $q \in \mathbb{Q}$ with $|r - q| < 2^{-p}$

**Examples.** Rational numbers, limits of computable Cauchy sequences $\pi$, $e$, …

**Definition.** A real function $f: \mathbb{R} \to \mathbb{R}$ is computable if there exists a Turing machine that computes $f(x)$ with an oracle for $x$.

**Examples.** Polynomials, trigonometric functions, …

**Counter-examples.** $x=0$, $\lceil x \rceil$ are not computable (undecidable on $x=0.000\ldots$) discontinuous functions

Analog encoding $e(w)$ of decision problems by $f$: accept $w$ if $f(e(w)) \geq 1$ reject if $\leq -1$
Analog Computer? Differential Analyzer [Bush 1931]

Underlying principles: Lord Kelvin, 1876
First ever built: Vannevar Bush, MIT, 1931

Applications: from gunfire control up to aircraft design
- Intensively used by the U.S. and Japanese armies during world war II
- Electronic versions from late 40s, used until 70s
General Purpose Analog Computer [Shannon 1941]

Shannon’s formalization of the Differential Analyser by GPAC circuits
A time function if GPAC-generated if it is the output of some unit of a GPAC circuit built from:
1. Constant unit
2. Sum unit
3. Product unit
4. Integral \( \int x \, dy \) unit

What does this GPAC circuit compute?

\[
\begin{align*}
y_1 &= \frac{dy}{dt} \\
\frac{dy_1}{dt} &= -y = y'' \\
\text{if } y(0) &= 1, \ y_1(0) = 0 \\
y(t) &= \cos(t) \quad y_1(t) = \sin(t)
\end{align*}
\]
CRN Implementation of GPAC Units

Mass action law kinetics reaction network with output concentration stabilizing on the result of the operation applied to the input concentrations

Positive constant units: molecular concentrations

Product unit \( z = x \cdot y \)   

\[
\begin{align*}
x + y & \quad \rightarrow \quad x + y + z \\
k.x.y & \\
z & \quad \rightarrow \quad _-
\end{align*}
\]

\[
\frac{dz}{dt} = k(xy - z) \quad \text{when } z = x \cdot y
\]

Sum unit \( z = x + y \)

\[
\begin{align*}
x & \rightarrow \quad x + z \\
k.x & \\
y & \rightarrow \quad y + z \\
k.y & \\
z & \rightarrow \quad _-
\end{align*}
\]

\[
\frac{dz}{dt} = k(x + y - z) \quad \text{when } z = x + y
\]

Time integral \( z = \int x \, dt \) unit

\[
\begin{align*}
x & \rightarrow \quad x + z \\
k & \\
\frac{dz}{dt} & = x \\
z & \rightarrow \quad \int_0^T x \, dt
\end{align*}
\]
Polynomial ODE Initial Value Problems (PIVP)

Graça and Costa 2003’s formalization of Shannon’s GPAC

Definition. A real time function \( f: \mathbb{R}_+ \to \mathbb{R} \) is GPAC-generable iff there exist a vector of polynomials \( p \in \mathbb{R}^n[\mathbb{R}^n] \) and of initial values \( y(0) \in \mathbb{R}^n \) and a solution function \( y: \mathbb{R}_+ \to \mathbb{R}^n \) such that \( y'(t) = p(y(t)) \) and \( f(t) = y_1(t) \)

Closure properties:
\[ f + g, f - g, f \cdot g, 1/f, f \circ g, \text{ s.t. } y' = f(y) \text{ are GPAC-generable if } f, g \text{ are.} \]

A GPAC-generated function must be analytic (i.e. convergent power series)

Famous analytic non-GPAC-generable functions [Shannon 41]
• Euler’s Gamma function \( \Gamma(x) = \int_0^\infty t^{x-1}e^{-t}dt \) [Hölder1887]
• Riemann’s Zeta function \( \zeta(x) = \sum_{k=0}^\infty \frac{1}{k^x} \) [Hilbert]
PIVP-Computable Functions $f(x)$

**Definition.** [Graça Costa 03 J. Complexity] A real function $f: \mathbb{R} \to \mathbb{R}$ is **PIVP-computable** if there exists vectors of polynomials $p \in \mathbb{R}^n[\mathbb{R}^n]$ and $q \in \mathbb{R}^n[\mathbb{R}]$ and a function $y: \mathbb{R}^n \to \mathbb{R}^n$ such that $y'(t) = p(y(t))$, $y(0) = q(x)$ and $|y_1(t) - f(x)| < y_2(t)$ with $y_2(t) \geq 0$ decreasing for $t>1$ and $\lim_{t \to \infty} y_2(t) = 0$.

**Example.** $\cos(4)$

**Theorem (analog characterization of Turing computability).**

[Bournez Campagnolo Graça Hainry 07 J. Complex]]

A real function is **computable (by Turing machine)** iff it is **PIVP-computable**.
Analog characterization of Ptime

Time in ODE is a bad measure of complexity
• Exponential speedup by changing time variable \( t' = e^t \)
• But price to pay in the amplitude of \( t' \)

A computational complexity measure should combine time and space-amplitude
• length in the \( n \) dimensions of the trajectory to compute the result

**Theorem** [Pouly PhD thesis 2015, Bournez Graca Pouly 16 ICALP]
A real function is computable in \( P \) iff it is PIVP-computable with a trajectory of polynomial length (i.e. polynomial time and polynomial amplitude)
Lemma (positive systems) Any PIVP-computable function can be encoded by a PIVP of double dimension on $\mathbb{R}^+$, preserving polynomial length complexity.

**Proof.** Encode $y_i \in \mathbb{R}$ by $y^-_i, y^+_i \in \mathbb{R}^+$ such that $y_i = y^+_i - y^-_i$ at each time (encoding used in [Oishi Klavins 2011] for linear I/O systems)

Let $p_i(y^+_1, y^-_1, \ldots, y^+_n, y^-_n) = p_i[y = y^+_i - y^-_i]$ and $p_i = p^+_i - p^-_i$

$y^+_i = q^+_i - f_i y^+_i y^-_i$ \quad $y^+_i(0) = \max(0, y_i(0))$

$y^-_i = q^-_i - f_i y^+_i y^-_i$ \quad $y^-_i(0) = \max(0, -y_i(0))$

Where $f_i = q^+_i + q^-_i$ are positive coefficient polynomials $f_i \geq \max(q^+_i, q^-_i)$

- Fast annihilation reactions: $\quad y^+_i + y^-_i \xrightarrow{f_i} -$ 
- n-ary catalytic synthesis reactions for each monomial $m^+_{i,j}$ in $p^+_i$, $m^-_{i,j}$ in $p^-_i$:
  
  $M_{i,j}^+ \xrightarrow{m^+_{i,j}} y^+_i + M_{i,j}$
  
  $M_{i,j}^- \xrightarrow{m^-_{i,j}} y^+_i + M_{i,j}$
Turing Completeness of Continuous CRNs 2/3

Lemma (quadratic systems) [Carothers Parker Sochacki Warne 2005]
Any PIVP can be encoded by a PIVP of degree $\leq 2$.

Proof. Introduce variable $v_{i_1,\ldots,i_n}$ for each possible monomial $y_1^{i_1}\ldots y_n^{i_n}$

We have $y_1 = v_{1,0,\ldots,0}$, $y_2 = v_{0,1,0,\ldots,0}$, ...

$y'_i$ is of degree one in $v_{i_1,\ldots,i_n}$

$v'_{i_1,\ldots,i_n} = \sum_{k=1}^{n} i_k v_{i_1,\ldots,i_{k-1},i_k,i_{k+1},\ldots,i_n}$ $y'_k$ is of degree at most 2.

i.e. trade high dimension for low degrees.

(yet naïve algorithm of exponential complexity)
Theorem (Turing completeness of continuous CRNs) [F Le Guludec Bournez Pouly CMSB 2017]

Any computable function over the reals can be computed by a continuous CRN over a finite set of molecular species (no polymerization, no locations)

Proof: By previous lemmas, any PIVP-computable function can be encoded by a PIVP of degree at most 2 with positive variables. A positive PIVP of degree at most 2 can be represented by an elementary CRN with at most 2 reactants per reaction.

In this view, the (protein) concentrations are the information carriers.

The programs of a cell are implicitly defined by the set of all possible reactions with the proteins encoded in its genome and the chemicals of the environment.

Program change is determined by gene expression (= metaprogram).

In this view, programming becomes a natural science
Theorem (abstract CRN)
A real function is computable (respectively in polynomial time) if and only if it is computable by a system of elementary reactions of the form

\[ \_ \rightarrow z \quad \text{or} \quad x \rightarrow x+z \quad \text{or} \quad x+y \rightarrow x+y+z \]

plus annihilation reactions \[ x+y \rightarrow \_ \] with mass action law kinetics (respectively with trajectories of polynomial length as a function of both the unary precision and the argument values).

Proof Close analysis of the encoding used in the lemmas (positive monomials)

Intermediate CRN: Instead of formal synthesis and degradation reactions, activation and deactivation reactions (e.g. phosphorylation, complexation)

Concrete CRN: catalogue of real enzymes [F. Molina’s Lab Sys2Diag, Montpellier] microreactors in DNA-free vesicles created by microfluidic device
Compilation of the Cosine(t) function

\texttt{biocham}: \texttt{compile\_from\_expression(cos,\textit{time},f)}.

\[
\begin{align*}
_ &= [z_p] \Rightarrow f_p. \quad z_m + z_p \Rightarrow _. \\
_ &= [z_m] \Rightarrow f_m. \quad f_m + f_p \Rightarrow _. \\
_ &= [f_m] \Rightarrow z_p. \\
_ &= [f_p] \Rightarrow z_m. \\
\text{present}(f_p,1).
\end{align*}
\]

\[
\begin{align*}
\frac{df_p}{dt} &= z_p - k \cdot f_m \cdot f_p \\
\frac{df_m}{dt} &= z_m - k \cdot f_m \cdot f_p \\
\frac{dz_p}{dt} &= f_m - k \cdot z_m \cdot z_p \\
\frac{dz_m}{dt} &= f_p - k \cdot z_m \cdot z_p \\
f_p(0) &= 1
\end{align*}
\]
Compilation of the Cosine(x) Function

biocham: present(x_p, 4).
biocham: compile_from_expression(cos,x,f).
  present(f_p, 1).
  _=[g_m]=>g_p.
  _=[x_p]=>g_p.
  _=[g_p]=>g_m.
  _=[x_m]=>g_m.
  _=[g_m+z_p]=>f_p.
  _=[x_p+z_m]=>f_p.
  _=[f_p+z_m]=>f_m.
  _=[g_p+z_p]=>f_m.

PIVP that generates f(g(t)) with \( \lim_{t \to \infty} g(t) = x \)

\[ g'(t) = x - g(t) \]
\[ g(t) = x + (x_0 - x)e^{-t} \]
Sigmoid Functions

Hyperbolic tangent
\[ \frac{d(HT)}{dt} = 1 - HT^2 \]

Logistic
\[ \frac{d(S)}{dt} = S - S^2 \]

Arc tangent
\[ \frac{d(T)}{dt} = 1 \]
\[ \frac{d(AT)}{dt} = \frac{1}{1+T^2} \]

Hill functions order 1, 2, 5
\[ \frac{d(H1)}{dt} = NH1^2 \]
\[ \frac{d(NH1)}{dt} = -NH1^2 \]
\[ \frac{d(H2)}{dt} = 2T*NH2^2 \]
\[ \frac{d(NH2)}{dt} = -2T*NH2^2 \]
\[ \frac{d(H5)}{dt} = 5T^4*NH5^2 \]
\[ \frac{d(NH5)}{dt} = -5T^4*NH5^2 \]
Logical Gates

**And**  \( C = A \land B \) \hspace{1cm} A+B => C \hspace{1cm} \lfloor C \rfloor = \min(\lfloor A \rfloor, \lfloor B \rfloor)\

**Or** \( C = A \lor B \) \hspace{1cm} A => C \hspace{1cm} B => C \hspace{1cm} \lfloor C \rfloor = \lfloor A \rfloor + \lfloor B \rfloor\

**Not** \( C = \neg A \) \hspace{1cm} C+A => _ \hspace{1cm} \lfloor C \rfloor = \max(\lfloor C_0 \rfloor - \lfloor A \rfloor, 0)\)
Computer-Aided Biochemical Programming of Synthetic Micro-reactors as Diagnostic Devices

Alexis Courbet\textsuperscript{1}, Patrick Amar\textsuperscript{2}, François Fages\textsuperscript{3}, Eric Renard\textsuperscript{4}, Franck Molina\textsuperscript{1}

\textsuperscript{1} Sys2diag UMR9005 CNRS/ALCEDIAG, Montpellier
\textsuperscript{2} LRI, Université Paris Sud - UMR CNRS 8623, Orsay
\textsuperscript{3} http://lifeware.inria.fr, Inria Saclay IdF, Palaiseau
\textsuperscript{4} INSERM 1411, Montpellier University Hospital
Protosensor CRN Design Workflow

Biomolecular problem to solve
- Pathology
- Molecular Biomarkers

Abstract logic function
- Temporal logic specifications

Biochemical programming
- Discrete & insulated microscale entities

Microfluidic assembly
- Molecular input signals

Functional protosensor
- Output signal

Automated design & implementation
- HSim: Realistic model prediction
  Hybrid entity centered/SSA automaton and ODE simulator
- BIOCHAM: Optimization & Model checking
  Sensitivity/Robustness analysis
  Temporal logic specifications

Silicell Maker
- Automated implementation of logic function from parts and devices mined in biological networks
Diabetes Differential Diagnostic Algorithm

Acute diabetes complications?

- Hypoglycemia?
  - Yes: Hypoglycemic
  - No: Glucosuria?
    - Yes: Lactaturia?
      - Yes: Ethanol?
        - Yes: Ethylism
        - No: Hyperosmolar Non-Ketotic
      - No: Lactic acidosis
    - No: Diabetic Ketoacidosis

- Lactaturia?
  - Yes: Lactic acidosis
  - No: Hyperosmolar Non-Ketotic

Screening for diabetes

- Glycosuria?
  - Yes: Urinary NOx?
    - Yes: Early onset of diabetes
    - No: No
  - No: No

GluONe

- Glucose
- Acetone
- Out1: 0
- Out2: 1

LacOH

- Lactate
- Ethanol
- Out1: 0
- Out3: 1

GluNOx

- NOx
- Glucose
- Out1: 0
- Out4: 1

JLLL'50 Roscoff, Mars 2019
François Fages
Reactions for Implementing Logical Gates

**And**  \( C = A \land B \)  
\( A + B \Rightarrow C \)  
\([C] = \min([A],[B])\)

**Or**  \( C = A \lor B \)  
\( A \Rightarrow C \)  
\( B \Rightarrow C \)  
\([C] = [A]+[B]\)

**Not**  \( C = \neg A \)  
\( C + A \Rightarrow \_ \)  
\([C] = \max([C_0]-[A], 0)\)
Microfluidic Assembly and Validation in Human Urine

GT Bioss, Marseille 2018
Sequentiality and Iteration

Division\((A, B)\)
\[
\begin{align*}
\text{begin} \\
01 & \text{ while } A \geq B \\
02 & \quad A := A - B \\
03 & \quad Q := Q + 1 \\
04 & \quad R := A \\
\text{end}
\end{align*}
\]

[Huang Jiang Huang Cheng 2012 ICCAD]
[Huang Huang Chiang Jiang F 2013 IWBDA]
Cell Division Cycle Program

while true {growing; replication; verification; mitosis}

→ compilation of *sequentiality* and *loops* with program control variables
→ 50 reactions
→ 13 variables

Cyclins D, E, A, B appear as necessary markers for implementing sequentiality
Conclusion

• Binary reaction systems over a finite set of molecules (without polymerization) are Turing-complete under the differential semantics
  – PIVP definition of computable function
  – Notion of computational complexity as trajectory length of stabilizing PIVPs

• Biochemical compiler of real functions (in Biocham modeling software)
  – Input: Function specification by PIVP, mixed digital-analog program
  – Output: system of binary reactions with mass action law kinetics
  – Exact characterization of the result for an ideal fluid implementation

• Comparison to natural circuits
  – The natural MAPK program implements an analog-digital converter (sigmoid Hill5)
  – Different from generated CRN for Hill5 but similar complexity

• Alternative design by artificial evolution: CRN ↔ Function
  – Artificial evolution of CRNs
  – Nature algorithms for learning [Valliant 2013]