3rd International BIOMICS Workshop, 2016
University of Passau, Germany

BIOMICS: Biological and Mathematical Basis of Interaction Computing

Project funded by the European Commission
Information and Communication Technologies

Collaborative Project, Grant no. 318202
Table of Contents

1 Introduction

Introductory Session
2 BIOMICS and Physics & Biology by Paolo Dini & Alastair Munro 4
3 BIOMICS and Mathematics by Gábor Horváth 13
4 BIOMICS and Automata by Chrystopher L. Nehaniv 24
5 BIOMICS and Specifications by Daniel Schreckling 36

Physics and Biology Session
6 Equilibrium, the Balance of natural Fallacy, and Dynamical Stability by Paolo Dini and Alastair Munro 40
7 Dynamical Stability, Hamilton-Jacobi Theory and the Quasi-Potential by Paolo Dini, Ágnes Bonivárt and Chrystopher Nehaniv 47
8 On Modelling Stochastic Reaction Kinetics: The XARNES Method by Zoran Konkoli 59

Mathematics and Automata Session
9 Lie symmetry analysis of the equation  \( \ddot{u} = f_0(u) + f_1(u)\dot{u} \) by Tamás Milkovszki 86
10 Holonomy Distribution and Degree of Metrizability by Zoltan Muzsnay 90
11 The Maximal Defect k-Subgroups of Semigroups of Graphs and Digraphs by Karoly Podoski 95
12 How can we find ordinary differential equations having a given a symmetry group by Ágota Figula 122
13 Automata and Specifications – Coupling as Colimit by Fariba Karimi and Chrystopher L. Nehaniv 148

Appendix
14 How to avoid Petri net idiosyncrasies when modeling computational systems by Egon Berger (Abstract) 171
15 BSL: aCoreASM Modification for the BIOMICS Project by Eric Rothstein Morris and Daniel Schreckling 172
16 LIFE: Load Balancing Inspired by Filament Structures by Daniel Schreckling, Eric Rothstein Morris and Chrystopher L. Nehaniv 177
## 1 Introduction

The Final Biomics Workshop hosted by the University of Passau took place 8-10 February 2016 and was well-attended by project partners and external guests. After an introductory presentation about the University of Passau by Professor Joachim Posegga (pictured), there were four high-level overview presentations on the four main areas of project research: biology and physics, mathematics, automata, and specification languages.

The flow of ideas in the project started many years ago with the observation that biochemical sub-cellular processes are able to construct order through the random interaction of their components, at different length and time scales. We postulated that this ability may be formalizable in mathematical terms from the point of view of (non-linear) dynamical systems. Because automata can be seen as a discrete form of dynamical systems, we then postulated that these ‘self-organized interactions’ could be expressed in computer science primitives or as interacting computational entities. Although we did not quite crack ‘the secret of life’, we feel that this idea is still valid and worth pursuing.

Some of us had underestimated the degree of difficulty of the mathematical challenges involved, resulting in a very intense and creative effort in each discipline of the project, and a very intense effort at understanding each other’s epistemological machinery. As is clear from the presentations and the papers in these proceedings, we were slightly more successful in the former than in the latter. An even bigger challenge has been to communicate effectively across the theoretical/applied divide, but in this the Passau team especially have achieved some very impressive results by extending the Abstract State Machines framework to Abstract State Interaction Machines.

Finally, a less explicit but no less important objective was to build a strong collaborative team of people who enjoy working together in spite of very large disciplinary, geographical, and cultural differences. In this third objective the BIOMICS project has been very successful. We were fortunate that Prof Zoran Konkoli from Chalmers University attended the workshop and gave a very interesting talk on the chemical master equation for stochastic models of biochemical systems with low particle numbers.
We organized the proceedings as a collection of presentation slides hoping that this would make the ideas presented more accessible to a wider interdisciplinary audience. Full details of the research presented can be found in the references provided by each presentation and in the project’s deliverables (www.biomicsproject.eu). In the appendix we also provide the abstract of a paper by Prof Egon Börger of the University of Pisa that has been accepted for publication and the written version of two papers that were presented at the workshop.

I hope that the excitement we felt in working on these topics and the fun we had working together comes across in these proceedings, and we thank the University of Passau for organizing and hosting a very enjoyable and interesting workshop.

Paolo Dini

BIOMICS Coordinator, on behalf of the BIOMICS Consortium

(Left to right) Zoltán Muzsnay, Eric Rothstein Morris, Pablo Andres Martinez, Chrystopher L. Nehaniv, Daniel Schreckling, Gábor Horváth, Zoran Konkoli, Tamás Milkovszki, Thomas Heistracher, Ágnes Bonivárt, Fariba Karimi, Paolo Dini, Ágota Figula
2- BIOMICS and Physics & Biology by Paolo Dini & Alastair Munro

BIOMICS and Physics & Biology

Paolo Dini
University of Hertfordshire

Alastair Munro
University of Dundee

Final BIOMICS Workshop
Passau, 8-9 February 2016

Linear Computing vs. Interaction Computing

- No clock: events driven by external triggers
- Many parallel and interdependent algorithms
- Interaction model is recursive: nested & hierarchical
- Memory is distributed
Open Systems, and Self-Organization as the Fall Towards Equilibrium

• The ‘self’ in self-organization indicates a spontaneous time evolution towards greater order.

• In physical systems spontaneous time evolution can only happen if a system is approaching or “falling towards” (thermodynamic) equilibrium.

• A closed system approaching equilibrium (minimum energy and/or maximum entropy) will eventually stop.

• Hence, in biology equilibrium is death, and must be avoided as long as possible – however, the fall towards equilibrium remains essential to self-organization.

• Therefore, biological organisms must be open to a flow of energy and information (i.e. food) that keeps them away from equilibrium even while they continue falling towards it!

• The broader picture is that this open process is happening on a wide staircase that starts at absolute zero and ends at the temperature inside stars. Biology is near the absolute zero end, where crystallization tends to win out over entropy.
Why Simplistic?

- In early 2012 we postulated that self-organization could be understood and modelled based on non-linear interactions and algebraic structure (e.g. symmetries or invariant quantities) of the systems in question.

- Although symmetries and algebraic structure still seem necessary, we are beginning to suspect they are not sufficient.

- So we are starting to think about 'constructive' models as well.

  These considerations apply to both biology and computation

- Furthermore, the temperature of computational systems is zero, in the sense that if they are not ‘pushed’ nothing happens. So thermodynamic analogies like the fall towards equilibrium and random mixing have to be made with care and may apply only in some limit or in a degenerate sense.

Models of Computation

(Theory catching up with practice...)

- Communication-only models
- Concurrency theory
  Communications before or after computations, between components, or between system and environment
  Pi Calculus
  etc

- Turing's 'Choice Machine' (1936)
  Communications happen during computation

- Turing machine
  • Closed-box transformations
  • Algorithms as function evaluations
  • Isolated computations
  • Internal time

- Computation-only model
‘Constructive’ Computational and Mathematical Models in the Context of Abstraction/Representation Theory

‘Physical computing is the use of a physical system to predict the outcome of an abstract [time] evolution’

• Normally, the computational model of an abstract evolution is a finite-state machine with a fixed number of states.

• However, if we think of ontogenetic processes like morphogenesis or even just ‘simple’ metabolism, clearly a fixed number of states is not sufficient.

• Furthermore, modern web-based computational architectures cannot be modelled with a single Turing Machine.

• BIOMICS is trying to develop the computational model – and the mathematical model above that – that corresponds to ‘constructive’ and interacting biological processes (and that may better support distributed architectures).

• The problem is that we don’t have a precedent for either kind of model, we have to invent a new computational model and some new mathematics above that.

A “Sufficiently Commuting” Diagram

Computations as mathematical objects

\[ m_p \overset{C(m_p)}{\longrightarrow} m_p \approx m'_p \]

Representation relative to theory \( T \) (e.g. classical, quantum, etc)

\[ \overset{\mathcal{R}_T}{\longrightarrow} \]

Physical device evolving under the laws of physics

Abstract

Physical

\[ p \overset{H(p)}{\longrightarrow} p' \]

A Computation

Mathematical model in initial state

Computational or machine model

\[ m_s \overset{E}{\longrightarrow} m_p \approx m'_p \]

Mathematical model in final state = ‘result’

\[ \overset{\mathcal{R}_T}{\longrightarrow} \]

\[ \overset{\mathcal{E}}{\longrightarrow} \]

\[ p \overset{H(p)}{\longrightarrow} p' \]

BIOMICS Computational Framework

*Starting with the Abstraction/Representation Theory perspective*

Mathematical Model

Instantiates into

Computational Model

Von Neumann Machine

BIOMICS Computational Framework

*Adding an emulation layer in place of actual natural computation*

Mathematical Model

Instantiates into

Computational Model

Implements

Interaction Machine

Emulates

One of More Von Neumann Machines
BIOMICS Computational Framework

Abstract View: Mathematical Model → Computational Model → Interaction Machine

Run-Time: Mathematical Model → Computational Model → Running Instance

Design-Time: BIOMICS Specification (Language) → System of ASM: Open to External Environment and Users

Biophysical Models of Interaction

Engineering Models of Interaction

One or More Von Neumann Machines
BIOMICS (Original) Research Framework

- Biology & Mathematics
  - Cellular pathways as biochemical dynamical systems
    - Evolution (Phylogeny) (Slow)
    - Genotype-phenotype map (Morphogenesis) (Ontogeny) (Fast)
    - Selection pressure or DNA specification

- Mathematics & Computer Science
  - Algebraic properties of automata structure
    - Lie group analysis, Petri net discretisation & holonomy decomposition
    - Category theory transformation
    - Bottom-up self-organisation (Fast)

- Computer Science & Software Engineering
  - Existing computational examples
  - Software engineering process (Slow)
  - Dynamic instantiation of Cloud services (Fast)

Interaction Machine Architecture

- Holonomy decomposition
- Category theory transformation

Still-Open Problems

- Formalization of stability: equilibrium, steady state, meta-stability, dynamical stability
- Relation of stability to epigenetics
- Nested structure of ecosystems: common language & language primitives
- Variable state space and constructive models
BIOMICS and Mathematics

Gábor Horváth (UD)

Ágota Figula, Tamás Mikovszki, Zoltán Muzsnay, Károly Podoski, Eszter Gselmann, Zoltán Halasi

3rd BIOMICS Workshop
Passau, 8 February 2016

Continuous vs discrete

Biological systems

Differential Equations
  \[ \text{Symmetries} \]
  \[ \text{Lie groups} \]

Finite automata
  \[ \text{Semigroups} \]
  \[ \text{Finite Simple Groups of Lie type} \]

Continuous
  \[ \text{Discrete} \]
Continuous vs discrete

Biological systems

Differential Equations

Symmetries

Lie groups

Differential Equations

Symmetries

Lie groups

Infinitesimal

Lie algebras

Continuous

Discrete

Finite automata

Semigroups

Finite Simple Groups of Lie type

Chevalley correspondence

???
Continuous vs discrete

Biological systems

Differential Equations
- Symmetries
  - real
  - Infinitesimal
  - real
  - Lie algebras

Finite automata
- Semigroups
  - complex
  - Lie algebras
  - Ch.
  - Finite Simple Groups of Lie type
Continuous vs discrete

Biological systems

Differential Equations

Symmetries

real

Lie groups

Infinitesimal

real

Lie algebras

???

simple complex

Lie algebras

Ch.

Finite Simple Groups of Lie type

Finite automata

Semigroups

Continuous

Discrete

Biomics
Lie symmetry group of 1st order ODEs

**Theorem**

Let \( I \subset \mathbb{R} \) be a nonempty open interval and \( M \subset \mathbb{R}^n \) be an open set, \( v \in C^1(I \times M) \) be a function for which there exists a continuous function \( L: I \rightarrow [0, +\infty) \) such that

\[
\|v(t,x_1) - v(t,x_2)\| \leq L(t) \|x_1 - x_2\| \quad (t \in I, x_1, x_2 \in \mathbb{R}^n).
\]

Then the Lie symmetry group of the equation

\[
\dot{x} = v(t,x)
\]

is isomorphic to

\[
G = \{ (t,x) \mapsto (f(t,x),g(x)) \mid f: I \times M \rightarrow I, g: M \rightarrow M \},
\]

where \((f,g): I \times M \rightarrow I \times M\) is a diffeomorphism.

**Problem:** G is HUGE. Even all diffeomorphisms of \( \mathbb{R}^n \) are unknown.
FitzHugh–Nagumo model

nerve impulse transmission

\[ \dot{v} = f(v) - w, \]
\[ \dot{w} = bv - \gamma w, \]

equivalent to 2nd order ODE

\[ \ddot{u} = (f(u) - bu) + \frac{\partial}{\partial t} (f(u) - u), \]

\[ \Rightarrow \text{Tamás Milkovszki} \]

\[ \ddot{u} = f_0(u) + f_1(u) \dot{u}. \]

Help: Lie bracket condition for 2nd order ODEs, Zoltán Muzsnay

Second order ODE: submanifold of the 2-jet space

Second order system

1. \[ \ddot{y}^j = \omega^j(x, y, \dot{y}) \]
2. \[ \Sigma^j(x, y, \dot{y}, \dot{\dot{y}}) := \ddot{y}^j - \omega^j(x, y, \dot{y}) \Rightarrow \Sigma^j(x, y, \dot{y}, \dot{\dot{y}}) = 0, \]

Solution

\[ \frac{d^2 y^j}{dx^2}(x) = \omega^j(x, y(x), \dot{y}(x)), \]
\[ \Sigma^j(x, y^j(x), \frac{dy^j}{dx}(x), \frac{d^2 y^j}{dx^2}(x)) = 0, \]
\[ j_2(\gamma) \subset \Sigma. \]
Lie bracket condition and the LSC

Spray: \( S^1 = \frac{\partial}{\partial x} + \dot{y}^i \frac{\partial}{\partial y^j} + \omega^i(x, y, \dot{y}) \frac{\partial}{\partial y^j} \)

Curve of the spray: \( \gamma: \mathbb{R} \rightarrow \mathbb{R}^n, \quad S^1(\dot{\gamma}) = \dot{\gamma} \)

\[
\begin{array}{ccc}
\mathbb{R} & \xrightarrow{\gamma} & \mathbb{R}^n \\
\downarrow & & \downarrow \\
J_2(\mathbb{R}^n) & \xrightarrow{\gamma} & \gamma \\
\downarrow & & \downarrow \\
\pi & \xrightarrow{\gamma} & \mathbb{R} \\
\downarrow & & \downarrow \\
J_1(\mathbb{R}^n) & \xrightarrow{\gamma} & \gamma \\
\downarrow & & \downarrow \\
S^1 & \xrightarrow{\gamma} & \gamma \\
\end{array}
\]

Proposition

The vector field \( X = \xi \frac{\partial}{\partial x} + \eta^i \frac{\partial}{\partial y^i} \) is an infinitesimal symmetry if and only if

\[
[S^1, X^1 - \xi S^1] = 0,
\]

where \( X^1 = \xi \frac{\partial}{\partial x} + \eta^i \frac{\partial}{\partial y^i} + (D\eta^i - \dot{y}^i D\xi) \frac{\partial}{\partial y^j} \).

Continuous vs discrete

- Biological systems
  - Continuous
  - Discrete

- Finite automata
  - Semigroups
  - Finite Simple Groups of Lie type

- Differential Equations
  - Symmetries
    - Simple real Lie groups
    - Simple complex Lie algebras

- Simple real Lie algebras
  - Ch.
  - Ch. groups of Lie type

\( \Rightarrow \) find differential equations for Lie groups
Differential equation associated to Lie groups and their tangential lie algebras

\[ G \text{ Lie group, } g \text{ Lie algebra } \quad \Rightarrow \quad \text{ODE (} \Sigma = 0 \text{)} \]

- \( \Sigma = 0 \) is a \( k \)th order ODE and \( X \in g \), then \( X^{(k)}(\Sigma) \equiv 0 \).
- \( g \ni \{ X_1, \ldots, X_r \} \)

\[ X^{(k)}_i = \phi_i(x, y) \frac{\partial}{\partial x} + \eta_i(x, y) \frac{\partial}{\partial y} + \eta^{(1)}_i \frac{\partial}{\partial y(1)} + \cdots + \eta^{(k)}_i \frac{\partial}{\partial y(k)} \]

- \( f \) ODE, \( M = \left( X^{(k)}_i \right)^T, i = 1, \ldots, r, k = r - 2 \)
- \( g \) is a subalgebra of the symmetry Lie algebra of \( f \) iff

\[ \left( \frac{\partial f}{\partial x} \quad \frac{\partial f}{\partial y} \quad \frac{\partial f}{\partial y'} \quad \cdots \quad \frac{\partial f}{\partial y^{(m)}} \right) \cdot M = (0 \quad \ldots \quad 0) \iff \det M = 0 . \]

- generalization \ldots Ágota Figula

\( \mathfrak{sl}_2(\mathbb{R}) \)

\[ X_1 = \frac{\partial}{\partial x} + \frac{\partial}{\partial y}, \quad X_2 = x \frac{\partial}{\partial x} + y \frac{\partial}{\partial y}, \quad X_3 = x^2 \frac{\partial}{\partial x} + y^2 \frac{\partial}{\partial y}, \]

\[ y^{(2)} = -\frac{2(y')^2 + y' + c (y')^3}{x - y}, \]

which is 2nd order, BUT

\[ X_1 = \frac{\partial}{\partial y}, \quad X_2 = y \frac{\partial}{\partial y}, \quad X_3 = y^2 \frac{\partial}{\partial y}, \]

\[ y^{(3)} - \frac{3(y^{(2)})^2}{2y'} - y'f(x) = 0, \]

there exists NO second order !!!
$so_3(\mathbb{R})$

\[ X_1 = (1 + x^2) \frac{\partial}{\partial x} + xy \frac{\partial}{\partial y}, \quad X_2 = xy \frac{\partial}{\partial x} + (1 + y^2) \frac{\partial}{\partial y}, \]
\[ X_3 = y \frac{\partial}{\partial x} - x \frac{\partial}{\partial y}, \]
\[ y^{(2)} = c \left( \frac{1 + y^2 - 2xy' + (1 + x^2)(y')^2}{1 + x^2 + y^2} \right)^{3/2}, \quad c \in \mathbb{R} \]

- There exists no solvable 2-dimensional subalgebra \( \implies \) even though we know a 3-dimensional Lie symmetry algebra, we may not be able to solve the ODE!
- Biomics is looking for simple Lie algebras. Helpful for solving ODEs are solvable Lie algebras. \( \implies \) hard to find a working example.

Rhodes’ conjecture

digraph $\Gamma$, edge $(x, y) \longrightarrow$ function $T_{x,y}: V \rightarrow V$
\[ T_{x,y}(v) = \begin{cases} 
    y & \text{if } v = x, \\
    v & \text{otherwise.} 
\end{cases} \]

The transformation semigroup of $\Gamma$:
\[ S(\Gamma) = \langle T_{x,y} \mid (x, y) \text{ is an edge of } \Gamma \rangle. \]

Full invariant of graphs
Rhodes’ conjecture

\[ S(\Gamma) = \langle T_{x,y} \mid (x, y) \text{ is an edge of } \Gamma \rangle. \]

defect \( k \)-subgroup: permutations moving \(|V| - k\) points

Conjecture (Rhodes)

- **defect 1**: product of cyclic, alternating, symmetric groups
- **defect 2**: product of alternating, symmetric groups
- **defect \( k \geq 3 \)**: product of symmetric groups

completely solved \( \implies \) Károly Podoski

Conclusion

- Not only isomorphism class but also presentation is important.
- For possible (biological) application of Chevalley correspondence, one needs to look for simple real Lie algebras.
- Coordinate free LSC condition is generalizable, gives insight.
- Knowing subgroups of Lie symmetry group of 1st order ODEs should give more insight into this symmetry group.
- Rewrite 1st order system to higher order system
- Old conjectures can be solved

AND...
Conclusion

- Not only isomorphism class but also presentation is important.
- For possible (biological) application of Chevalley correspondence, one needs to look for simple real Lie algebras.
- Coordinate free LSC condition is generalizable, gives insight.
- Knowing subgroups of Lie symmetry group of 1st order ODEs should give more insight into this symmetry group.
- Rewrite 1st order system to higher order system
- Old conjectures can be solved

AND...

- Tamás Milkovszki (Monday, 15:10)
- Zoltán Muzsnay (Monday, 16:30)
- Ágota Figula (Monday, 17:20)
- Károly Podoski (Tuesday 9:30)
4. BIOMICS and Automata by Chrystopher L. Nehaniv
Symmetry Groups

- Transformation Semigroup of Automaton
- Substructure permutation groups: natural subsystems (local reversibility for subset of states that is the image of a projection)
- Automorphism groups & Subgroups of Endomorphisms Monoids: Internal automorphism groups of automata come from external ones \( \rightarrow \) decomposition using symmetries (Egri-Nagy & Nehaniv 2015)
- Simple Non-Abelian Groups (SNA\&G) arising in models of biological systems \( \rightarrow \) Functional Completeness & Finitary Computational Universality (Maurer-Rhodes 1965; Horváth-Nehaniv 2015)
- Groups arising from Automorphisms Systems of Differential Equations (Lie) & Equations (Galois).
- Morphisms of these Equation Systems (Karimi-Nehaniv) \( \rightarrow \) Endomorphisms Monoids of Systems of Equations and their Subgroups.
Table 1. States in a Boolean network model of the lac operon. The states are defined by Boolean combinations of the presence/activity versus absence/irreversibility of biochemical components: L, lactose; A, allolactose; Z, an isomer of lactose; Op, the repressor molecule; ZYA, the structural genes for the enzymes needed for lactose metabolism. In the Boolean network, 1 means that the molecule is present/active or the gene is expressed; 0 is for the absence/inactivity. States are given by binary column vectors, e.g., 1000 corresponds to the state L,A, where L and A have value 1 and Op and ZYA have value 0. States to the left of the vertical separating line correspond to metabolic states in which lactose is present.

Lac Operon in E. coli

natural subsystems

weak control hierarchy

(computed for examples using our SgpDec package for the computer algebra system GAP)
Weak Control Hierarchy of the Lac Operon

Emden–Meyerhof pathway and Krebs cycle. (a) Enzymes involved in the reactions as numbered in the diagram are: (1) pyruvate dehydrogenase, (2) citrate synthase, (3) aconitase, (4) and (5) isocitrate dehydrogenase, (6) α-ketoglutarate dehydrogenase, (7) succinyl-CoA synthetase, (8) succinyl-CoA synthetase, (9) fumarase and (10) malate dehydrogenase. (b) Reaction graph and finite-state automation model $K_C$ of the Krebs cycle (see text). Substrates correspond to states of the automation and if acted on by the relevant coenzyme are transformed, e.g. pyruvic acid ($x_1$) interacting with NAD and CoASH ($e_5$) is transformed into acetyl-CoA ($x_3$). This is denoted $\delta(x_1,e_5) = x_3$. Note that if the coenzymes have no effect on a substrate then no arrow is shown, e.g. $\delta(x_2,e_4) = x_2$ as oxaloacetate acid ($e_4$) is not transformed by interaction with COP ($e_6$). The presentation is "event-based" or "perturbation-based" and time does not appear explicitly in the model. Reactions shown run to completion, and thus all basic transitions are idempotent (i.e. $\delta(x_3,e_5) = \delta(x_3,e_5) = x_3$) always holds in a reaction graph. (Models from Rhodes [11], ch. 6, Part 1.)
Reaction Graphs
Labelled directed graphs with no two-edge sequence with a repeated label. (Reactions run to completion.)

Special case: Each edge with a distinct label (Other have semigroups that are substructures of these.)

Their natural subsystems are characterised now in BIOMICS solving a conjecture of Rhodes (see K. Podoski talk tomorrow: defect-k groups of digraphs)
Two-molecule tracking Krebs Cycle

Krebs cycle two-molecule automation model. States are given by one to two metabolites, while transitions are given by coenzymes and energy carriers (indicating input of pyruvic acid (P)) using the same notation as in figure 4(b). Only transitions which change the state are shown, e.g., (x y z)-arrow, as acetyl CoA and citric acid in the presence of CoA (with appropriate enzymes) are transformed to citric acid (then isocitric acid), whereas (x y z)-arrow as acetyl CoA and isocitric acid are anabolized in the presence of NADP (and enzyme uncoupled). This version of the Krebs cycle allows explicitly how to and where in the cycle to insert the two molecules. Whether two molecules are inserted simultaneously or not to the cycle is rather irrelevant for this simplified model.

Natural Subsystems and Weak Control Hierarchy: 2-molecule Krebs Cycle Model
Some Categories & Functors for Systems Biology

Natural Subsystems and Weak control hierarchy for the p53- mdm2 case with binary values for the levels of M (mdm2), C (p53- mdm2 complex) (p53) and R (p53*).

\[ N \] acting on 4 points

\[ M \] acting on 4 points

\[ C \] acting on 4 points

\[ R \] acting on 4 points

\[ t \] acting on 2 points

**Natural Subsystems and Weak Control Hierarchy for the p53- mdm2 petri net with capacities for M=2, C=1, P=1 and R=1.**

\[ S_1 = (\alpha_1, \alpha_2) \] acting on 6 points

\[ A_1 = (\beta_1, \beta_2, \beta_3) \] acting on 5 points

\[ S_2 = (\gamma_1, \gamma_2) \] acting on 4 points

\[ A_2 = (\delta_1, \delta_2, \delta_3, \delta_4) \] acting on 5 points

\[ S_3 = (\epsilon_1, \epsilon_2) \] acting on 3 points

\[ S_4 = (\zeta_1, \zeta_2, \zeta_3, \zeta_4) \] acting on 4 points

\[ A_3 = (\eta_1, \eta_2, \eta_3, \eta_4) \] acting on 5 points

\[ S_5 = (\kappa_1, \kappa_2) \] acting on 3 points

\[ S_6 = (\lambda_1, \lambda_2, \lambda_3) \] acting on 3 points

\[ S_7 = (\mu_1, \mu_2, \mu_3, \mu_4) \] acting on 4 points

\[ S_8 = (\nu_1, \nu_2, \nu_3, \nu_4) \] acting on 4 points

\[ S_9 = (\xi_1, \xi_2, \xi_3) \] acting on 3 points

\[ S_{10} = (\zeta_1, \zeta_2, \zeta_3) \] acting on 3 points

**3rd International BIOMICS Workshop, Passau 2016**
General (Static) Automata Network

Formulation of the base \( n \) expansion of natural numbers as a dynamic cascade automata network. A cascade of modulo \( n \) counters \( C_n \) gives the usual base \( n \) expansion of integers with a dynamically changing number of digits. Taking \( m = 10 \), the top cascade holds three-digit base \( n \) numerical representations, while the bottom cascade holds four-digit base \( n \) numerical representations. The number of digits may increase if a carry results in a non-zero value beyond the leftmost digit, while if subtraction would result in the leftmost digits becoming zero then at the same time the leftmost automata for these digits are removed, resulting in a cascade of fewer automata. (Example due to C. L. Nehaniv and E. Rothstein Morris.)
Weak Control Hierarchy gives an ordering of Natural Subsystems from which the Automaton can be synthesized in a feedforward cascade.

This gives a coordinate system on the automaton under study: successive approximation, projection, coordinate computation in using permutation groups and constants - generalizing the base n expansions of numbers

This Weak Control Decomposition corresponds to the holonomy decomposition proof of the Krohn-Rhodes Decomposition Theorem (wreath product decomposition)

Moreover, one can use a dynamically growing cascade instead of a static one.

Cell cycle Petri net & Automaton (capacity 1)

(a) Petri net model of the cell cycle based on a differential equation model of reactions as described by Tyson.
(b) Finite six-state automaton associated to this Petri net whose state set consists of all configurations reachable from the initial configuration with one token at the place representing MPF (active or inactive).
Regenerate and Re-Use despite Irreversibility

Use in Interaction Machines for Regeneration

Multiple Copies (e.g. offspring, ensembles of biosystems): Living things reverse irreversibility (!)

Re-use weak control / holonomy coordinates

Despite eventual irreversible decay (after ‘transients’ to ergodic mode), i.e. death entire structure recurs in offspring / copies

Interaction Machines

A level n+1 interaction machine is a dynamic automata network of interaction machines of level n or below. The entire structure and each constituent component updates its state based on local conditions and environmental input. Based on interactions and local state the network may change its interconnection topology and how components affect each other (i.e. the functions that locally determine inputs to constituent components), as well as changing the number and type of components dynamically. Interaction machines and their constituents of level 1 or above may grow or remove components as needed in response to interactions.
5. BIOMICS and Specifications by Daniel Schreckling

BIOMICS and Specifications

Daniel Schreckling
IT-Security Group
University of Passau
Passau, Germany

3rd BIOMICS Summer Workshop
February 2016

Research Agenda Towards Specifications

- Definition of an interaction machine model (WP3)
- Develop language which allows specification of computation through behaviour (WP2, WP3, WP4)
  - What are feasible language primitives?
  - Are there specific logics specifying the behaviour?
  - How to obtain usable specifications and their tools?
- Machine and specifications reflect construction principles known from bio-chemical systems (WP1, WP2, WP3)
- Implementation of an interaction machine realizing specifications (WP4, WP5)
Discovery and Derivation of Category Theoretic Tools

- Categories and functors
- Adjunctions their realizations and categories
- F-(Co)algebras and their final/initial elements
- Bisimulation and Bisimilarity
- Monads their specific categories (Kleisli and Eilenberg-Moore categories) and applications

Algebraic/Categorical Structures in Simple Automata

- Analyzing typical CS problems with Holonomy Decomposition
  - n-Queens Problem and its solutions
  - Consumer-producer problem
  - Various Petri-Net models
- Find mapping of properties to decomposition properties
Co-Algebraic/Behavioural Specification of ICM

- General idea
  - Static structures and properties can be described algebraically
  - Dynamics and behaviour of systems can be described coalgebraically
  - Use category theory to map (co)algebras of system to (co)algebraic properties of the language
  - Find a specification language able to express static and coalgebraic properties

- Reality
  - Algebraic and coalgebraic insights remain limited
  - Derivation of concrete language remains underspecified task
  - Consequence: Adopt general purpose language

Mapping ASMs to Coalgebra

- ASMs are feasible to model systems we know (ASM thesis)
  - combine developments of formal logic in decades
  - introduce algorithms on real world objects
  - abstract states allow for appropriate abstraction level

- Start coalgebraic modelling of ASMs
  - Develop functor signatures able to express ASM machines
  - Specify categories for specific ASMs
  - Derive final $F$-coalgebras specifying the behaviour of such machines
  - Show refinements of ASMs and their corresponding functors
  - Apply to simple Shabbat-Lift and its refinements

- Result: ASMs are (co)algebraic and appear to be feasible for BIOMICS
Extending Existing ASM Model to Model ICM

- Collect high-level requirements
- Discover properties existing models similar to interaction computing
  - Distributed algorithms
  - P systems
  - Rewriting systems
  - Population protocols
- Introduce concurrency with scheduling policies
- Define Biomics Specification Language (BSL) able to specify ICM behaviour

Implementing the ICM with a BSL runtime

- CoreASM realizes ASM specifications
- Extension of CoreASM can execute BSL specifications
- Implement scheduling policies for agents (an ICM instance)
- Specify and run population protocols using BSL
- Unify some insights from BIOMICS in a single simulation
Equilibrium, the Balance of Nature Fallacy, and Dynamical Stability

Paolo Dini
School of Computer Science
University of Hertfordshire, Hatfield, UK

Alastair Munro
Division of Cancer Research
University of Dundee, Dundee, UK

Abstract

'The Balance of Nature is a theory that proposes that ecological systems are usually in a stable equilibrium, which is to say that a small change in some particular parameter will be corrected by some negative feedback that will bring the parameter back to its original “point of balance” with the rest of the system.' As suggested by Adam Curtis in his documentary ‘The Use and Abuse of Vegetational Concepts’, Part 2 of the series, All Watched Over by Machines of Loving Grace, this theory was dominant until approximately 1980, when fieldwork, undertaken as part of a project led by George van Dyne, suggested that no such balance existed. The irony is that van Dyne was in fact trying to do the opposite, to confirm the Balance of Nature theory. Ecosystems, it seems, will not return to some ‘equilibrium’ when disturbed by small- or large-scale disasters: new configurations will ensue.

Ecological population models are among the most well-studied dynamical systems, usually modelled with sets of coupled, non-linear differential equations. In BIOMICS, we are interested in studying cell metabolic and regulatory pathways because they appear to exhibit a form of stability under external perturbations that is generally associated with the state of ‘health’ of the organism. We wish to understand this form of stability and see whether it can be formalised as ‘self-organising behaviour’ of systems in general and, in particular, of computational systems. The biochemical systems of the cell are also modelled with systems of coupled, non-linear differential equations. Although the empirical evidence that ecosystems are not ‘stable’ in any conventional sense of the term is by now compelling, so is the behavioural distinction between, for example, a healthy and a cancerous cell. Therefore, it would appear that the concept of stability is rather subtle and needs to be handled differently for biological systems in different contexts (e.g. over different spatial and temporal scales).

As a motivational effort towards more mathematical modelling and formalisation, this short paper provides a brief historical literature review of the concept of stability in dynamical systems theory and proposes a conceptual model of a generalization thereof dubbed ‘dynamical stability’ in BIOMICS that appears to be necessary for self-organization. This has interesting resonance with one of the paradoxes of evolutionary biology: if the laws of thermodynamics indicate that entropy should increase over time, then why does evolution produce increased organisation and complexity? Perhaps ‘dynamical stability’ is part of the answer.

2 http://en.wikipedia.org/wiki/All_Watched_Over_by_Machines_of_Loving_Grace_%28TV_series%29

1

40
Equilibrium, the Balance of Nature Fallacy and Dynamical Stability

Paolo Dini  
University of Hertfordshire  

Alastair Munro  
University of Dundee

Equilibrium

- ‘Equilibrium’ and ‘steady state’ are often used interchangeably, but they are not necessarily synonymous.

- The configuration space of a dynamical system is the space defined by treating the dependent variables as independent dimensions. A solution of the system then becomes a curve in this space parametrized by time.

- In dynamical systems theory, an equilibrium point in configuration space is a point at which the rate of change of the problem variables with respect to time is zero.

- An isolated dissipative system that is started away from equilibrium will eventually “fall” back to the equilibrium state.

- The portion of the solution of an isolated system that is a function of time but that is approaching equilibrium is called the transient response (shown in red):
Stable Equilibrium and Oscillations

- If we characterize the state of a system as a point in configuration space, the following quotation from Crawford (1965) seems helpful:

  *The world is full of things that move. Their motions can broadly be categorized into two classes, according to whether the thing that is moving stays near one place or travels from one place to another.*

- Oscillations belong to the first class, and are characterized by three things:
  - the existence of an equilibrium state
  - the presence of a return force when the system is displaced from equilibrium
  - the presence of an inertia that causes the system to overshoot equilibrium

- The presence of a return force makes the equilibrium point *stable*.

Steady State

- **Steady state** indicates more generally the condition where the transient has been damped by friction. In a driven (i.e. not isolated) oscillator, the steady-state part of the solution is still a function of time, but has become periodic.

- The same idea applies in higher dimensions.

![Graph of transient (approx.) and steady-state = periodic](image)
Dynamical Equilibrium

- On the other hand, Newton’s law can also be regarded as a condition of “dynamical equilibrium”: the motion of an accelerating mass could therefore be seen in dynamical equilibrium simply because it is consistent with

\[ F = ma \]

- So one needs to pay attention to the context in which these terms are used
Active Feedback Control

System and Biological Stability

- A system may have one or more stable equilibrium points, but its behaviour may still be unstable.

- A dynamical system is considered to be **stable** if external perturbations cause oscillations whose amplitude does not increase over time. It is **unstable** otherwise.

- If we regard biological organisms as complex dynamical systems and if we associate health with equilibrium, then according to this definition their ability to heal themselves would seem to make healthy organisms ‘stable’.

- However, systems that we can model mathematically and that are stable are not capable of exhibiting complex behaviour in any way resembling biological organisms.

- So, “biological stability” must be a more complex concept and mathematical property.

- Stability as phenomenological illusion: a system may appear to be stable on the outside but may rely on complex and locally unstable interactions on the inside. This is getting closer to self-organization through interactions.
The Balance of Nature Fallacy

(Curtis 2011)

- Until approximately 1980, ecology as a science was based on the assumption that ecosystems are stable: if a hurricane, or a fire, or an earthquake disturbed an ecosystem, the assumption was that after some time the ecosystem would regain its original equilibrium state: Nature was assumed to be ‘in balance’.

- In the late 1970s, fieldwork performed by the ecologist George van Dyne, however, showed fairly convincingly that ecosystems are not in balance: a large perturbation causes an ecosystem to drift to a different configuration, potentially very different from the original.

- The irony is that van Dyne was in fact trying to do the opposite, to confirm the Balance of Nature theory.

- It is possible that the Balance of Nature fallacy originates from the fact that in (bio)chemical systems we cannot properly talk about an inertia. Hence the importance of time-delays in oscillating biochemical systems.

- Be that as it may, does this result from ecology have some implications for BIOMICS???


Generalizations

- When a theory does not fit anymore, we (sometimes) generalize.

- A first possibility is to say that – rather than a single equilibrium point – complex non-linear systems have multiple equilibria. Systems with two or more stable equilibria are called meta-stable.

- The transitions between different meta-stable states result from external inputs, i.e. from interactions of the system with its environment.

- Biological systems appear to transition between different meta-stable states continuously, in real time, as a result of the continuous interactions between external inputs and internal components, at different length and time scales.

- We have tentatively called such continuously varying trajectory of meta-stable states dynamical stability.

- Dynamical stability may be a conserved quantity or a function of conserved quantities. Hence our effort to use Lie symmetries to guide the analysis.

- More recently, we have begun to suspect that a probabilistic/stochastic approach may be necessary to explore meta-stable systems through the ‘quasi-potential’.
Dynamical Stability: *Formalize This!*

Static stability

Dynamical stability:
Low-dimensional or "discrete"

High-dimensional or continuous

Ping-pong ball balanced on air-dryer

Leonard Everett, uconn.edu


www.greylabynth.com
Dynamical Stability, Hamilton-Jacobi Theory and the Quasi-Potential

Paolo Dini, Ágnes Bonivárt and Chrystopher Nehaniv
University of Hertfordshire

Overview

- Probability theory basics
- Master Equation (Fokker-Planck, Langevin, Smoluchowski-Kramer)
- Hamilton-Jacobi theory
- Large deviation theory
- Quasi-potential
Meta-Stability of Dynamical Systems

- The RHS of a system of ODEs such as
  \[
  \dot{x} = f_1(x, y) \\
  \dot{y} = f_2(x, y)
  \]
  can be seen as a vector field in phase space ($\mathbb{R}^2$ in this case).

- Setting this field equal to zero yields $N \geq 0$ attracting fixed points of the phase flow.

- If $N > 1$, these points are called **meta-stable states** of the system, because the system may transition between them if subject to external perturbations.

- This is a physically plausible statement, but it is not reflected by the mathematical model above. To account for external perturbations and general noise we introduce a probability-based stochastic model.

Stochastic Variables and Stochastic Processes

(The following several slides are based on Van Kampen, N G (2007). *Stochastic Processes in Physics and Chemistry*, North-Holland)

- A **stochastic variable** $X$ is a variable that can take on values from a given range with a given probability distribution.

- A **stochastic process** is a stochastic variable that is defined as a function of $X$ and $t$:
  \[
  Y_X(t) : \mathbb{R}^2 \rightarrow \mathbb{R}, \quad Y_X(t) = f(X, t).
  \]

- A **sample function** or realization is obtained by inserting a particular value $x \in X$, thereby obtaining an ordinary function of time:
  \[
  Y_x(t) = f(x, t).
  \]
Ensemble of Sample Functions

- We define a probability density $P_X(x)$ over the values of $X$, such that the ensemble average or expectation value of the stochastic process is a function of time given by

$$\langle Y(t) \rangle = \int Y_X(t) P_X(x) dx.$$ 

Moments and Stationary Processes

- The $n$th moment is defined as

$$\langle Y(t_1) Y(t_2) \cdots Y(t_n) \rangle = \int Y_X(t_1) Y_X(t_2) \cdots Y_X(t_n) P_X(x) dx,$$

where $t_1, t_2, \ldots, t_n$ are not necessarily all different.

- A stochastic process is stationary if the moments are not affected by a shift in time: $\langle Y(t_1 + \tau) Y(t_2 + \tau) \cdots Y(t_n + \tau) \rangle = \langle Y(t_1) Y(t_2) \cdots Y(t_n) \rangle$. 

Multivariate Distributions

- Let $X$ be a random variable with $r$ components $X = (X_1, \cdots, X_r)$, so $X \in \mathbb{R}^r$.

- Its probability density is a function $P_r(x_1, \cdots, x_r) : \mathbb{R}^r \to U$, with $U = [0, 1] \subset \mathbb{R}$.

- $P_r$ is called the joint probability distribution (JPD) of the $r$ (component) variables.

- Take a subset $(X_1, X_2, \cdots, X_s) \in \mathbb{R}^s$, with $s < r$. The probability that they have values $(x_1, x_2, \cdots, x_s)$ is called the marginal distribution:

$$P_s(x_1, x_2, \cdots, x_s) = \int P_r(x_1, x_2, \cdots, x_r)dx_{s+1}dx_{s+2}\cdots dx_r,$$

so distribution over first $s$, dependence on $s\cdots r$ integrated away

Bayes’s Rule

- If we pick a specific set of values for $(x_{s+1}, x_{s+2}, \cdots, x_r)$, the JPD of the rest is called the conditional probability:

$$P_{s|r-s}(x_1, \cdots, x_s|x_{s+1}, \cdots, x_r).$$

- It is not hard to see that

$$P_r(x_1, \cdots, x_r) = P_{r-s}(x_{s+1}, \cdots, x_r)P_{s|r-s}(x_1, \cdots, x_s|x_{s+1}, \cdots, x_r).$$

- This is called Bayes’s rule, usually expressed as

$$P_{s|r-s}(x_1, \cdots, x_s|x_{s+1}, \cdots, x_r) = \frac{P_r(x_1, \cdots, x_r)}{P_{r-s}(x_{s+1}, \cdots, x_r)}.$$
Markov Processes

- In a Markov processes the transition probability between two states depends only on the starting state of the transition and not on the history of the process that took place before that point:
  
  \[ P_{1|n-1}(y_n,t_n|y_1,t_1; y_2,t_2; \ldots; y_{n-1},t_{n-1}) = P_{1|1}(y_n,t_n|y_{n-1},t_{n-1}). \]

- The Chapman-Kolmogorov equation (CKE) is derived by making use of Bayes's rule and of the Markov property:
  
  \[
  P_3(y_1,t_1; y_2,t_2; y_3,t_3) = \frac{P_2(y_1,t_1; y_2,t_2) P_1(y_3,t_3|y_1,t_1; y_2,t_2)}{P_1(y_1,t_1) P_1(y_2,t_2|y_1,t_1) P_1(y_3,t_3|y_2,t_2)}.
  \]

CKE Equation and Semigroups

- Now integrate both sides over all possible values of \( y_2 \), such that \( y_2 \) drops out of the LHS:
  
  \[
  P_2(y_1,t_1; y_3,t_3) = P_1(y_1,t_1) \int P_1(y_2,t_2|y_1,t_1) P_1(y_3,t_3|y_2,t_2) dy_2.
  \]

- Dividing by \( P_1(y_1,t_1) \) and applying Bayes's rule we obtain the CKE:
  
  \[
  P_{1|1}(y_3,t_3|y_1,t_1) = \int P_{1|1}(y_3,t_3|y_2,t_2) P_{1|1}(y_2,t_2|y_1,t_1) dy_2.
  \]

- Interestingly, this equation can be seen as the closure condition for the contraction semigroup of a suitably defined family of Markov processes. This is discussed more fully in Chapter 1 of Freidlin & Wentzell (1998), who build on E Dynkin's work on Markov processes.
Homogeneous Markov Processes

- A non-stationary Markov process is homogeneous if it starts at a specific time $t_0$ (and therefore it is non-stationary) but, after that, its transition probability depends only on the time difference between states:

$$P_{t_1|t_2}(y_2,t_2|y_1,t_1) = T_{t_2-t_1}(y_2|y_1), \quad \tau = t_2 - t_1.$$ 

- Such that the CKE becomes

$$T_{t_2-t_1}(y_3|y_1) = \int T_{t_2-t_1}(y_3|y_2)T_{t_2-t_1}(y_2|y_1)\,dy_2.$$ 

Probabilistic Ontological Expansion

- In order to work with these homogeneous processes, van Kampen (implicitly) introduces an apparently contradictory definition for 'sub-ensemble' which has important ontological implications.

- He 'extracts a sub-ensemble', by picking a particular probability distribution $P(y_0)$, where $y_0$ ranges over all the values of $Y$, as before.

- By defining a probability distribution over $Y$ the ontology of this mathematical quantity is expanded: the stochastic variable $Y$ is not only a determinisitic set of possible values, but also an infinite set of possible probability distributions over this set. (negative & imaginary numbers, etc).

- There are many problems in physics where a deterministic description does not lead to a solvable mathematical problem, but where if the problem is recast in terms of probability a solution (for the latter) can be obtained.
Continuous Master Equation

- After a couple of pages of work, letting $\tau' \to 0$ in the CKE we get

$$\frac{\partial T_{\tau}(y_3|y_1)}{\partial \tau} = \int \left[ W(y_3|y_2)T_{\tau}(y_2|y_1) - W(y_2|y_3)T_{\tau}(y_3|y_1) \right] dy_2,$$

where $W(y_i|y_j)$ – which comes from the linear term of the Taylor expansion – is the transition probability per unit time between states $i$ and $j$.

- Letting now $y = y_3$ and $y' = y_2$, we obtain the continuous Master Equation:

$$\frac{\partial P(y,t)}{\partial t} = \int \left[ W(y|y')P(y',t) - W(y'|y)P(y,t) \right] dy'$$

with initial condition $P(y_1,t_1) = \delta(t - t_1)$.

- This PDE is an equation for the transition probability from $(y_1,t_1)$ to any $(y,t)$.

Discrete Master Equation

- In discrete form, for a (probabilistic) automaton of $n = \{1, \cdots, N\}$ states,

$$\frac{dp_n(t)}{dt} = \sum_{n'} \left[ W_{nn'} p_{n'}(t) - W_{n'n} p_n(t) \right]$$

- For each member of the sum, the first term is the transition probability flux from state $n'$ into state $n$, weighted by the probability that the automaton is in state $n'$. The second term is the flux from state $n$ out to state $n'$, weighted by the probability that the automaton is in state $n$.

- Conservation of probability equation, similar to fluid mechanics! 😊

- For a finite-state automaton, there is a Master Equation for each state.

- The probability flux factors $W$ must come from an independent condition, for each pair of states and in each transition direction between them.

- Master Equation implies that the probability distribution over the set of states changes as a function of time.
Fokker-Planck Equation

- Loosely speaking, the Fokker-Planck equation can be considered a Taylor series approximation of the continuous Master Equation up to 2nd order:

\[
\frac{\partial P(y,t)}{\partial t} = -\frac{\partial}{\partial y}[A(y)P] + \frac{1}{2} \frac{\partial^2}{\partial y^2}[B(y)P].
\]

Or,

\[
\frac{\partial P(y,t)}{\partial t} = -\frac{\partial J(y,t)}{\partial y},
\]

where

\[
J = [A(y)P] - \frac{1}{2} \frac{\partial}{\partial y}[B(y)P]
\]

is the probability flux.

Langevin Equation

- van Kampen says that the Langevin equation for Brownian motion is mathematically equivalent to the Fokker-Plank equation.

- However, Langevin’s equation is essentially Newton’s law:

\[
\mu \ddot{q}^{\mu,\varepsilon} = \text{force term} + \text{fluctuating force term} - \text{friction force term}
\]

\[
\mu \ddot{q}^{\mu,\varepsilon} = F(q^{\mu,\varepsilon}) + \varepsilon \dot{\xi}(t) - \mu \dot{q}^{\mu,\varepsilon}
\]

- In the Smoluchowski-Kramer approximation, the inertia term is neglected, leading to the typical form of a stochastic differential equation (SDE):

\[
\dot{q}^{\varepsilon} = F(q^{\varepsilon}) + \varepsilon \dot{\xi}(t)
\]

- This has the same form as the metabolic systems of ODEs we are interested in, plus a fluctuating component.
Hamilton-Jacobi Theory

- Frictionless (Hamiltonian) dynamical systems follow trajectories in configuration space that minimize the action:

\[ S = \int L \, dt = \int \left[ KE(q(t)) - PE(q(t)) \right] \, dt, \]

where \( S \) is a solution to the Hamilton-Jacobi equation (HJE):

\[ H(q, \frac{\partial S}{\partial q}, t) + \frac{\partial S}{\partial t} = 0, \]

and \( H \) is the Hamiltonian function for the system.

- The HJE is a 1st-order PDE whose characteristics are solutions to the Hamilton equations of motion and the action

\[ \dot{p} = -\frac{\partial H}{\partial q}, \quad \dot{q} = \frac{\partial H}{\partial p}, \quad S = L = \sum q \dot{p} - H, \]

and \( p \) is the momentum.

Metabolic Systems

- Metabolic systems are not frictionless and are not Hamiltonian. Thus, the force field \( F(q) \) in general is not the negative gradient of a potential.

- Rather, it is assumed that it is possible to decompose it into a gradient term and a ‘remainder’ term:

\[ F(q) = -\nabla \tilde{U} + F_r, \]

where \( \tilde{U} \) is the quasi-potential function.

- Out of several possible decompositions, Zhou et al. (2012) find that a normal decomposition yields a \( \tilde{U} \) that satisfies a Lyapunov function’s conditions:

\[ F(q) = -\nabla U_{\text{norm}} + F_\perp, \]

- Quasi-potential is also assumed to related to the probability of transition between different meta-stable states.
Large Deviation Theory (very messy!)

- For SDEs, the most likely transition trajectories between two points lie in a small neighbourhood of the path $q$ minimizing the Freidlin and Wentzell action (which is based on the fluctuating component):
  \[ S_T(q) = \frac{1}{2} \int_0^T |\phi - F(\phi)|^2 dt, \]

- Because (1) the action is based on fluctuations and (2) probability of transitions are associated with the quasi-potential, the Freidlin-Wentzell action is identified with a potential: $S_T(\phi) \equiv V$.

- As a result, the HJE takes the form (I have not derived this yet!!):
  \[ \frac{1}{2} |\nabla V|^2 + (\nabla V, F) = 0 \]


HJE and Normal Decomposition

- Zhou et al. (2012) show that the change in Freidlin and Wentzell potential is twice the change in the normal quasi-potential:
  \[ V_{AB} = \frac{1}{2} \int_{t_A}^{t_B} |\phi - F(\phi)|^2 dt = 2(U_B^{\text{norm}} - U_A^{\text{norm}}) \]

- They then say (without deriving it) that the HJE for such systems is given by the orthogonality condition itself:
  \[ (-\nabla U^{\text{norm}}, F) = 0 \]
  \[ (-\nabla U^{\text{norm}}, F + \nabla U^{\text{norm}}) = 0 \]

- And proceed to claim that this condition is sufficient to solve for $U^{\text{norm}}$. ☺

Meta-Stability of First-Order Systems

- Given a system of ODEs

\[
\begin{align*}
\dot{x} &= -1 + 9x - 2x^3 + 9y - 2y^3 \\
y &= 1 - 11x + 2x^3 + 11y - 2y^3
\end{align*}
\]

- Dynamical systems may have more than one fixed point or stable state, shown as local minima of the quasi-potential in this figure:


Visualization of Normal Decomposition

Tentative Conclusions

- Assuming all the holes in the derivations can be plugged, the quasi-potential seems like an interesting concept.

- Combining the Master Equation with the quasi-potential seems to offer the beginnings of a formalism to connect biological behaviour to probabilistic finite-state automata through probability.

- Probability formalism seems amenable to the kind of mathematical analysis we have been developing (Lie symmetries, category theory, etc)
8 On Modelling Stochastic Reaction Kinetics: The XARNES Method by Zoran Konkoli

On modelling stochastic reaction kinetics: the XARNES method

BIOMICS Workshop
February 2016

Zoran Konkoli
Chalmers University of Technology
The Department of Microtechnology and Nanoscience --- MC2

Background

• All chemical reactions are intrinsically stochastic, but this feature does not always manifests itself. Most of the time, we can be happily oblivious to that fact.
• However, there are situations where noise cannot be ignored. Noise can render the equations of classical chemical kinetics unreliable.
• Such “noisier” systems require a special attention: we know how to model them in principle, but solving the mathematical models is a challenge.
• Examples: intracellular reaction networks (e.g. gene expression networks, or complex formation), exciton dynamics, matter-anti matter annihilation in the early universe,...
A summary of the following publications


2012

2011

2010
- Konkoli, Z. A danger of low copy numbers for inferring incorrect cooperativity degree. Theoretical Biology and Medical Modelling, 7, 40 (2010).

2004

Outline

- What is the problem? What is so hard about modelling stochastic chemical reactions?
  - Noise can render any ODE based prediction qualitatively wrong (utterly unreliable)
- There are several noise types to consider.
  - spatial noise, temporal noise/intrinsic noise, spatio-temporal noise
  - fluctuation dominated kinetics
- Examples of models/systems that exhibit such behavior
  - the ABBA model (spatial noise)
  - living cell (temporal-noise/intrinsic noise)
- Moment closure methods
  - spatially extended systems (briefly)
  - well-mixed systems (complex formation)
What is the problem? Several noise types to consider

Spatial noise: geometry of noise

q: a microscopic concentration profile
**Reaction hole/cavity as spatial fluctuation**

Diffusion is slow in mixing particles. Once hole is created through reaction it takes time before this hole (cavity) is filled through diffusion.

(a) Reactants mixed well!

(b) Reactants are not mixed well!

(c) Reactants mixed well!

Cavity created through reaction

Takes a lot of time to fill the cavity by diffusion!

Mean field dynamics describes well mixed situation correctly, e.g. (a) and (c). It fails to describe the situation when reactants are not mixed (b). When reactants are distributed as in (b) one says that system exhibits fluctuation dominated kinetics.

Diffusion CONTROLLED reactions: If reaction rate A is much larger than diffusion rate D most of the time system spends in (b). When pair of particles disappears through reaction the diffusion decides how much time it will take to the next reaction. Thus diffusion controls kinetics.

**Reaction hole/cavity concept explained through diff. equation**

\[
\frac{\partial n(x,t)}{\partial t} = D \frac{\partial^2 n(x,t)}{\partial x^2}
\]

\[
\dot{n}(x,t) = D n''(x,t)
\]

\[
\dot{n}(k,t) = -Dk^2 n(k,t)
\]

Large scale fluctuations (small \(k\)) smear out much slower than small scale (large \(k\)) fluctuations

No change in concentration. Well mixed situation!

\(n(x_0, t) = 0 \Rightarrow \dot{n}(x_0, t) = 0\)

\(n''(x_0, t) > 0 \Rightarrow \dot{n}(x_0, t) > 0\)

\(n''(x_0, t) < 0 \Rightarrow \dot{n}(x_0, t) < 0\)

\(n(x_0, t+dt) > n(x_0, t)\): concentration grows at \(x_0\) due to inflow of neighbouring particles by diffusion

\(n(x_0, t+dt) < n(x_0, t)\): concentration drops at \(x_0\) due to outflow of particles by diffusion
Temporal noise

$q$ (an example): stochastic time series data (protein copy number)

Temporal noise (intrinsic noise, shot noise): very few particles in the system

We assume that particle positions are irrelevant.
The complexity comes from elsewhere:

$$P(n_1, n_2, \ldots, n_N; t)$$

$n_i$ = typically a very low integer

$N$ = typically a very large integer
Spatio-temporal noise

The five assumptions (traps) of “classical” biochemistry:

- Living has an infinite volume
- Number of reactants is infinite (very large)
- The cell interior is dilute
- Substrate concentrations are greater than enzyme concentrations
- The cytoplasm is well defined and homogeneous solution

Examples: spatial noise

**Examples of Diffusion Controlled Reactions and emergence of fluctuation dominated kinetics**

D-diffusion constant
λ,δ-reaction rates

most interesting for λ,δ >> D

\[
\begin{align*}
A + A & \xrightarrow[\lambda]{t} 0 \\
\dot{n}_A &= -\lambda n_A^2, \quad n_A(t) \sim (\lambda t)^{-1} \\
n_A(t) &\sim A(d)t^{-d/2} \\
A + B & \xrightarrow[\delta]{t} 0 \\
n_A(t) &= n_B(t) \sim A(n_0, d)t^{-d/4}
\end{align*}
\]
Steric Effects in Diffusion Controlled Reactions: ABBA model


\[
\begin{align*}
A + A & \xrightarrow{\lambda} 0 \\
A + B & \xrightarrow{\delta} 0 \\
B + B & \xrightarrow{\lambda} 0
\end{align*}
\]

\[
\begin{align*}
\frac{d}{2} & \quad \text{d/2} \\
\frac{d}{4} & \quad \text{d/4}
\end{align*}
\]

\[
\begin{align*}
& n_{A,B}(t) \sim A(d)t^{-d/2} \\
& n_A(t) = n_B(t) - A(n_0, d)t^{-d/4} \\
& n_A(t) \sim A(n_{0A}, n_{0B}, \lambda, \delta, d)t^{-d/2} \\
& n_B(t) \sim B(n_{0B}, n_{0B}, \lambda, \delta, d)t^{-d/2}
\end{align*}
\]

Examples: Temporal noise
About using the many body theory to understand the living cell


“A popular notion of complex systems is of very large numbers of simple and identical elements interacting to produce ‘complex’ behaviours. The reality of biological systems is somewhat different. Here large numbers of functionally diverse, and frequently multifunctional, sets of elements interact selectively and non-linearly to produce coherent rather than complex behaviours.”

There is a combinatorial explosion that we are not used to deal with

A typical situation: There are many types of molecules but they appear in low copy numbers

Technical/conceptual problems:
- Many particle labels (a very large number of field indices)
- Sparse configuration space
- Exponentially large master equation; \( P(n_1,n_2,n_3,\ldots,n_{20},\ldots) \)

Left: Many shapes but only few of each type.
Right: same as on the left but with clear functionality associated to each element: Giuseppe Arcimboldo, “Vertumnus”, 1590-1591: Rudolph II of Prague
For realistic biochemical networks the size of the master equation becomes an issue

The number of configurations becomes exponentially large (even with the well-mixed reaction volume assumption). It is hard to compute observables directly from the master equation.

\[ \vec{c} = (n_1, n_2, \ldots, n_i, \ldots, n_T) \]

\[ \langle f(\vec{c}) \rangle = \sum_{\vec{c}} f(\vec{c}) P(\vec{c}, t) \]

This sum is very hard to compute for large systems
An idea around the problem: moment closure methods

- The key idea: Derive equations of motion for observables of interest and bypass the master equation. Try to extend the usual ODE approach by using more variables/observables.
- Good observables? n-point functions, factorial moments

An ODE track: we hit a problem

- Equations of motion are not closed, they form an infinite hierarchy. The equations need to be closed (moment closure methods).
- Example: A+A reaction master equation:
  \[ \dot{P}(n, t) = \lambda \frac{(n+2)(n+1)}{2} P(n+2,t) - \lambda \frac{n(n-1)}{2} P(n,t); n = 0,1,2,\ldots, \infty \]
  \[ \frac{d}{dt} \langle n \rangle = \sum_n n \frac{d}{dt} P(n,t) = -\lambda \langle n(n-1) \rangle + \lambda \langle n^2 \rangle = -\lambda (n(n-1)) = -\lambda (n^2) + \lambda(n) \]
- The equation for the first order moment involves the second order moment!
  \[ \frac{d}{dt} \langle n \rangle = -\lambda_2 \langle n^2 \rangle + ODE_1; \lambda_2 \equiv \text{const} \lambda \]
- Likewise for the higher order moment equations:
  \[ \frac{d}{dt} \langle n^2 \rangle = -\lambda_3 \langle n^3 \rangle + ODE_2 + ODE_1; \lambda_3 = \text{const} \lambda \]
  \[ \frac{d}{dt} \langle n^3 \rangle = -\lambda_4 \langle n^4 \rangle + ODE_3 + ODE_2 + ODE_1; \lambda_4 = \text{const} \lambda \]
Quantifying the presence of noise?

- Weak noise: the following assumption can be used safely
  \[ \mu_2 = \langle n^2 \rangle \approx \langle n \rangle \langle n \rangle = \langle n \rangle^2 = \mu_1^2 \]
- Likewise for higher order moments
  \[ \mu_i = \langle n^i \rangle \approx \langle n \rangle^i = \mu_1^i \]
- In general, the pair correlation function determines:
  \[ \chi \equiv \frac{\langle n^2 \rangle}{\langle n \rangle^2} : \begin{cases} \approx 1; \text{noise is not important} \\ \neq 1; \text{noise is very important} \end{cases} \]

Meaning of mean field approximation: it implicitly assumes perfect mixing (the lowest order MCM)

Fundamental quantity is pair correlation function
\[ n(x,y,t) = n(x,t)n(y,t)\chi(x,y,t) \]

Mean field approximation
\[ \chi = 1 \Rightarrow n(x,y,t) \approx n(x,t)n(y,t) \]

Caution! By doing so one implicitly assumes that system is in the states (a) and (c) most of the time.

Modelling stochastic reaction kinetics: the XARNES method

An example of a moment closure method for well-mixed systems: XARNES

the X-level Approximation Reaction Noise Estimator method

- well mixed systems (living cell)
- many particle types (living cell)
- “automatization” (black-box software implementation) possible
- accuracy control available

The XARNES idea

To use correlation forms which are used in statistical physics to model spatially extended systems (the Kirkwood superposition approximation formalism).

Inspiration from:


Suitable for extending to d=0
The factorial moment as an observable

\[
\rho_{m} = \frac{n_1!}{(n_1-m_1)!} \frac{n_2!}{(n_2-m_2)!} \cdots \frac{n_l!}{(n_l-m_l)!} \frac{m_1!}{m_1!} \frac{m_2!}{m_2!} \cdots \frac{m_l!}{m_l!}
\]

\[
\rho_{(1,0,0)} = \langle n_1 \rangle = \sum_{n_1,n_2,n_3} n_1 P(n_1,n_2,n_3)
\]

m-vectors

\[
\rho_{(3,2,1)} = \langle n_1(n_1-1)(n_1-2)n_2(n_2-1)n_3 \rangle
\]

\[
= \sum_{n_1,n_2,n_3} \cdots P(n_1,n_2,n_3)
\]

XARNES with its X can do calculations quite a few, but one thing has not been seen: just what does X really mean?

A two type system example; \( P(n,m) \)

- moments with \( \xi \leq X \) kept
  (defines base moments)

- moments with \( \xi > X \) approximated

The meaning of \( \xi \)

<table>
<thead>
<tr>
<th>( \xi )</th>
<th>Factorial moments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( \rho_{(1,0)} = \langle n \rangle ), ( \rho_{(0,1)} = \langle m \rangle )</td>
</tr>
<tr>
<td>2</td>
<td>( \rho_{(2,0)} = \langle n(n-1) \rangle ), ( \rho_{(0,2)} = \langle m(m-1) \rangle ), ( \rho_{(1,1)} = \langle nm \rangle )</td>
</tr>
<tr>
<td>3</td>
<td>( \rho_{(3,0)} = \langle n(n-1)(n-2) \rangle ), ( \rho_{(2,1)} = \langle n(n-1)m \rangle ), ( \rho_{(1,2)} = \langle nm(m-1) \rangle ), ( \rho_{(0,3)} = \langle m(m-1)(m-2) \rangle )</td>
</tr>
</tbody>
</table>

...
Method description

\[
\frac{d}{dt} \rho_{\tilde{m}}(t) = \sum_{\alpha=1}^{R} \lambda_{\alpha} \sum_{\tilde{c} \in \Omega_R} \left( \begin{array}{c} \tilde{m} \\ \tilde{c} \end{array} \right) \Gamma_{\alpha}(\tilde{c}) \rho_{\tilde{m} - \tilde{c} + \tilde{u}_\alpha}(t)
\]

Equations for factorial moments form an infinite hierarchy of ODEs. A moment closure method is a recipe of decoupling the hierarchy.

\[
\Gamma_{\alpha}(\tilde{c}) = \left[ \left( \begin{array}{c} \tilde{u}_\alpha \\ \tilde{c} \end{array} \right) - \left( \begin{array}{c} \tilde{u}_\alpha \\ \tilde{c} \end{array} \right) \right] \frac{\tilde{c}!}{\tilde{u}_\alpha!}
\]

\[
\left( \begin{array}{c} \tilde{m}_1 \\ \tilde{m}_2 \end{array} \right) = \left( \begin{array}{ccc} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \end{array} \right) \cdots \left( \begin{array}{c} m_{1i} \\ m_{2i} \end{array} \right) \cdots \left( \begin{array}{c} m_{1T} \\ m_{2T} \end{array} \right)
\]

The mathematical foundation

The formula can be obtained by carefully compressing spatial degrees of freedom in the related spatially extended approach.

\[
\ln \nu_{\tilde{m}}(t) = \sum_{\tilde{m} \in \Omega_\xi} \left( \begin{array}{c} \tilde{m} \\ \tilde{m} \end{array} \right) w_{\tilde{m}}(t)
\]

\[
\frac{d}{dt} \rho_{\tilde{m}}(t) = \sum_{\alpha=1}^{R} \lambda_{\alpha} \sum_{\tilde{c} \in \Omega_R} \left( \begin{array}{c} \tilde{m} \\ \tilde{c} \end{array} \right) \Gamma_{\alpha}(\tilde{c}) \rho_{\tilde{m} - \tilde{c} + \tilde{u}_\alpha}(t)
\]
A technical impression of what one has to deal with: PARNES; X = 2(Pair)

\[ \langle \dot{a} \rangle \equiv \langle a \rangle^2 \chi_{AA}(t) \]
\[ \langle \dot{p} \rangle \equiv \langle p \rangle^2 \chi_{PP}(t) \]
\[ \langle \dot{a} \dot{p} \rangle \equiv \langle a \rangle \langle \dot{p} \rangle \chi_{AP}(t) \]
\[ \langle \dot{a}^2 \dot{p} \rangle \equiv \langle a \rangle^2 \langle \dot{p} \rangle \chi_{PP}(t)^2 \chi_{AP}(t) \]

\[ \frac{\partial}{\partial t} \chi_{AA} \equiv -\frac{\lambda}{(k-1)!} a^{k} \chi_{AA}^{k(k-1)} + \delta k p + j \]
\[ \frac{\partial}{\partial t} \chi_{PP} \equiv -\frac{2k \delta}{a} \chi_{PP}^{k} \end{array} \]
\[ \frac{\partial}{\partial t} \chi_{AP} \equiv \frac{2k \lambda}{a} \chi_{AP}^{k} \chi_{PP}^{k} - 2 \gamma \chi_{AP} \]

The standard ODE part

The extended ODE part

Accuracy tests (an example)


Factorial moments computed by using the XARNES with xi=1,2,... for a complex formation model (Poisson initial state). (1) Systematic accuracy improvement can be achieved. (2) Notice dramatic accuracy increase already at the second order (pair) level. Why is that?
A derivative matching procedure shows that the method should work extremely well for Poisson-like systems.

Konkoli, *Mathematical explanation of the predictive power of the X-level approach reaction noise estimator method.*


\[
\frac{d}{dt} \rho_{\bar{m}}(t) = \sum_{\alpha=1}^{R} \lambda_{\alpha} \sum_{\bar{c} \in \Omega_R} \left( \frac{\bar{m}}{\bar{c}} \right) \Gamma_{\alpha}(\bar{c}) \rho_{\bar{m}-\bar{c}+\bar{c}_{\alpha}}(t)
\]

\[
\ln \nu_{\bar{m}}(t) = \sum_{\bar{m} \in \Omega_{\bar{m}}} \left( \frac{\bar{m}}{\bar{m}} \right) w_{\bar{m}}(t) \quad \psi_{\bar{m}}(\bar{v}(t)) = \prod_{\bar{m} \in \Omega_{\bar{m}}} \nu_{\bar{m}}(t)^{\gamma_{\bar{m}}}
\]

**Few takeaway words about the method**

- The XARNES idea is base on the formalism of correlation forms (but adapted to study the living cell)

- XARNES = X level Approach Reaction Noise Estimator. A series of methods: X=S, X=P, X=T, ...

- Computational cost: extends the ODE’s for average particle numbers;
  \[ N \rightarrow N(S) + N^2(P) + N^3(T) + \ldots \]
Thank you!

An afterthought: There are many ways to make music:

Space: \( \langle a^+ (r_1) b^+ (r_2) \rangle \)

Zero dimensional field theory but with a large number of fields:
\[ \langle a^+ b^+ c^+ d^+ e^+ f^+ g^+ \ldots \rangle \]

Göteborgs symfoniker

Tamburitza orchestra from 1900 (USA)

Modeling of intracellular noise is very important

Noise can affect speed of biochemical reactions, average concentrations and flows in the living cell. It is, in fact, possible that the "taming" of noise in gene expression has been a major driving force in the evolution of extant intracellular feed-back systems.

Many proteins occur in low copy numbers

From: Taniguchi ... Xie; Science 329 (5991), 533 (2010). Average copy numbers of essential proteins (blue) and all proteins (pink).

Some proteins are maintained at low copy numbers: No wonder...
Just remember the central dogma the cell biology:
gene $\rightarrow$ mRNA $\rightarrow$ protein

only one for a given protein type
a few

Low copy numbers lead to noisy dynamics

intrinsic vs extrinsic noise

Noise cannot always average out and needs to be controlled by the cell. A failure to do so results in a cell disorder (disease).

Traces of fluctuation dominated kinetics

Figure 1: Result of Monte Carlo simulations in 1d for type (i) initial condition. A very large system is simulated on a lattice with $L = 10^4$ sites. Also, the initial number of particles $N_0(0) = N_0 = 5000$ is very large. Simulation starts from the largest possible density $n_0(0) = n$ particle/site. A and B particles have the same diffusion constant $D_A = D_B = 1$.  

Figure 2: Various situations which are simulated are shown. The three squares schematically depict various types of initial conditions from which simulation is started. (a) particles react when nearest-neighbors only. (b) Situation of dense packing with a large reaction range. It corresponds to situation of high packing which occurs in a cell environment. (c) It is unrealistic that particles can penetrate into each other midway between two extremes.

The XARNES ansatz is a multiplicative ansatz

Reason to be optimistic: (1) The coefficients do not depend on the form of a reaction model. (2) In the spatially extended world a multiplicative ansatz is usually more accurate than an additive one. A few reasons to worry: (1) But does the same hold in the well mixed regime, i.e. is the method accurate? (2) Can one improve accuracy by increasing the degree of the highest correlation form used?
Technical details

\[ \rho_{\bar{m}}(t_0) = \nu_{\hat{m}}(t_0) = \mu_{\hat{m}}^{\bar{m}} ; \bar{m} \in \Omega_\xi \]

The first and the second derivative match exactly when the initial pdf is the Poisson distr.

\[ \frac{d}{dt} \rho_{\bar{m}}(t) \bigg|_{t=t_0} = \frac{d}{dt} \nu_{\hat{m}}(t) \bigg|_{t=t_0} \]

\[ \frac{d^2}{dt^2} \rho_{\bar{m}}(t) \bigg|_{t=t_0} = \frac{d^2}{dt^2} \nu_{\hat{m}}(t) \bigg|_{t=t_0} \]

Apps: noise of complex formation kinetics

Why complexes? Because nobody has a clue how they work, and the (classical chemical) kinetics of their formation is much less understood than their function, not to mention the noise related issues...

Williamson, J.R.

*Cooperativity in macromolecular assembly.*

An interesting observation


“We also study the fluctuations in protein levels and find them to be significantly smaller in large complexes, and in the least abundant subunit of each complex.”
The key model: kA into P (well mixed)

We assume that particle positions are irrelevant. The complexity comes from elsewhere:
\[ P(n_1, n_2, \ldots, n_N, t) \]
\[ n_i = \text{typically a very low integer} \]
\[ N = \text{typically a very large integer} \]

Two results

- Noise reduction can be spontaneous, this is at least a possibility.
- The dose response curve should not be trusted for low copy numbers.
Noise reduction can be spontaneous


$\eta_{XY}(t) = \frac{\sigma_{XY}^2(t)}{\mu_A(t)\mu_P(t)}$

Model: Hill’s model noise characteristics (ligand binding, receptor activation)

$C_0 + hA \leftrightarrow C_h$

Exact dose response curve

PARNES dose response curve

Theoretical Biology and Medical Modelling 7, 40 (2010)
The simplest bistable model:

\[ Y \xrightarrow{\lambda_1} 2X \]
\[ 2X \xrightarrow{\lambda_2} X + Y \]
\[ X + Y \xrightarrow{\lambda_3} Y \]
\[ X \xrightarrow{\lambda_4} 0 \]

In the macroscopic (mean field) limit, the system has two stationary states. The first state has no particles in the system.

Wilhelm BMC Sys Bio 2009

Higher order correlation forms are needed to describe bistable systems.

Figure 11: Factorial moments computed by the use of the XARNES methods with \( \xi = 1, 2, \ldots, 4 \) for the smallest bistable model computed with the same choice of parameters as in Fig. 10. Higher order correlation forms are needed to describe bistable systems.
An example of an application to a more realistic model

A negative feedback loop model (bursty dynamics). The protein produced by a gene dimerizes and acts as a negative (inhibitory) transcription factor of the same gene. (Five particle types).


Non-trivial distribution function as a result of bursty dynamics

Long inactivity periods with few particles interrupted with periods of activity.

Figure 12: Simulation data for a small negative auto regulation gene expression network (please see the text). All simulation runs were done for for $0 \leq t \leq t_{\text{max}}$, where $t_{\text{max}} = 10s$ and the reaction rate parameters were chosen as discussed in the text. The histogram depicts the distribution of $P$ particles at $t = t_{\text{max}}$. Each simulation is started from the Poisson state with means equal to $\mu_i = (0.1, 0.1, 0.1, 0.1, 0.1)$ for $GP_2$, $G$, $R$, $P$, and $P_2$ particles respectively. To make the plot 100000 runs were used.
Skewness and kurtosis: to describe them accurately a higher order theory is needed

Figure 13: Skewness [panel (a)] and kurtosis [panel (b)] for the particle number distribution of the negative auto regulation model discussed in Fig. 12. Both were computed by using the XARNES method with $\xi = 1, 2, 3, \text{and} 4$. 

Zoran Konkoli: zoran@chalmers.se
9 Lie symmetry analysis of the equation $\ddot{u} = f_0(u) + f_1(u)\dot{u}$ by Tamás Milkovszki

Lie symmetry analysis of the equation $\ddot{u} = f_0(u) + f_1(u)\dot{u}$

Tamás Milkovszki

3rd international BIOMICS Workshop, Passau, February 08, 2016.

The research by the authors leading to these results was funded in part by the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 318202.

- FritzHugh-Nagumo model:
  \begin{align*}
  \dot{v} &= v(1 - v)(v - a) - w + l, \\
  \dot{w} &= bv - \gamma w,
  \end{align*}
  \hspace{1cm} (1, 2)

- Expressing $w$ from (1) and substituting into (2) we have
  \begin{align*}
  \ddot{u} &= f_0(u) + f_1(u)\dot{u}, \\
  \end{align*}
  \hspace{1cm} (3)

where:

\begin{align*}
  f_0(u) &= \gamma (-u^3 + u^2 - ua + u^2a) - bu \\
  f_1(u) &= -3u^2 + 2u(1 + a) - a - \gamma
  \end{align*}
We seek the infinitesimal generator of a symmetry in the form

\[ X = \xi(t, u) \frac{\partial}{\partial t} + \eta(t, u) \frac{\partial}{\partial u}. \]

The symmetry condition:

\[ [X^1 - \xi S^1, S^1] = 0, \]

where

\[ X^1 = \xi \frac{\partial}{\partial t} + \eta \frac{\partial}{\partial u} + (D\eta - \dot{u}D\xi) \frac{\partial}{\partial \dot{u}}, \]

\[ S^1 = \frac{\partial}{\partial t} + \dot{u} \frac{\partial}{\partial u} + (f_0(u) + f_1(u) \dot{u}) \frac{\partial}{\partial \dot{u}}. \]

The symmetry condition gives that

\[
\begin{align*}
(f_0' \eta + f_1 \eta_t - \eta_{tt} + 2 \xi_t f_0 - f_0 \eta_u) \\
+ (f_1' \eta + f_1 \xi_t - 2 \eta_{tu} + \xi_{tt} + 3 f_0 \xi_u) \cdot \dot{u} \\
+ (-\eta_{uu} + 2 \xi_{tu} + 2 f_1 \xi_u) \cdot (\dot{u})^2 \\
+ \xi_{uu} \cdot (\dot{u})^3 = 0.
\end{align*}
\]

Hence we obtain

\[
\begin{align*}
f_0'(u) \eta + f_1(u) \eta_t - \eta_{tt} + 2 \xi_t f_0(u) - f_0(u) \eta_u &= 0, \quad (4) \\
f_1'(u) \eta + f_1(u) \xi_t - 2 \eta_{tu} + \xi_{tt} + 3 f_0(u) \xi_u &= 0, \quad (5) \\
-\eta_{uu} + 2 \xi_{tu} + 2 f_1(u) \xi_u &= 0, \quad (6) \\
\xi_{uu} &= 0. \quad (7)
\end{align*}
\]
• We restrict our attention to the case $\xi_u(t, u) = 0$, $\eta_t(t, u) = 0$.

• $\xi_u(t, u) = 0 \Rightarrow \xi(t, u) = \xi(t)$,
  $\eta_t(t, u) = 0 \Rightarrow \eta(t, u) = \eta(u)$.

• From (6) we conclude that $\eta_{uu}(u) = 0$. So we obtain
  $$\eta(u) = au + b.$$ 

• If $f_0(u) \neq 0$, then differentiating (4) with respect to $t$ we have
  $$2\xi_{tt}(t)f_0(u) = 0 \Rightarrow \xi = ct + d.$$ 

• So (4) gives that
  $$f'_0(u)(au + b) = f_0(u)(a - 2c) \Rightarrow f_0(u) = A_1(au + b)^{-\frac{2c}{a} + 1},$$

• and from (5) we find that
  $$f'_1(u)\eta(u) + f_1(u)c = 0 \Rightarrow f_1(u) = B_1(au + b)^{-\frac{c}{a}}. \quad (8)$$

• If $f_1(u) = 0$, then (3) can be solved by elementary methods.

**Theorem**

Let $\mathcal{L}$ be the Lie subalgebra of all symmetries of $\ddot{u} = f_0(u) + f_1(u)\dot{u}$ defined by the equations $\xi_u = 0$, $\eta_t = 0$. If $f_0(u) = A_1(au + b)^{-\frac{2c}{a} + 1}$, $A_1 \neq 0$ and $f_1(u) = B_1(au + b)^{-\frac{c}{a}}$ then $\mathcal{L}$ is two dimensional and the generators of $\mathcal{L}$ are $Y_1 = \frac{\partial}{\partial t}$ and

$$Y_2 = t \frac{\partial}{\partial t} - \frac{f_1(u)}{f'_1(u)} \frac{\partial}{\partial u}.$$
If $f_0(u) = 0$, then differentiating (5) with respect to $u$ we obtain

$$f''_1(u)\eta + f'_1(u)\eta_u + f'_1(u)\xi_t = 0$$

(9)

Differentiating (9) with respect to $t$ we have

$$f'_1(u)\xi_{tt} = 0.$$  

(10)

Therefore $f'_1(u) = 0$ or $\xi_{tt} = 0$. If $f'_1(u) = 0$ then (3) takes the form $\ddot{u} = K\dot{u}$. If $\xi_{tt} = 0$ then $\xi = ct + d$ and we have that $f_1(u) = B_1(au + b)^{-\frac{c}{a}}$

Let us suppose that $f_0(u) = u^{1+2c}$, $f_1(u) = u^c$. Then (3) takes the form $u^{1+2c} + u^c \ddot{u}$. An equivalent first order system is

$$\dot{u} = uv,$$

$$\dot{v} = u^{2c} + u^c v - v^2.$$
Holonomy distribution and degree of metrizability of a SODE

Z. Muzsnay, S. Elgendi

Third BIOMICS Workshop
University of Passau, Germany

February 8 - 10, 2016.

Calculus of variations:

- scalar product \( \langle v, v \rangle = E(v) \)
  
  Riemann: \( \langle v, v \rangle_x \)
  
  Finsler: \( \langle v, v \rangle_{(x, y)} \)

- \( I[\gamma] = \int_{\gamma} E(\gamma, \dot{\gamma}) \)

\[
\frac{d}{dt} \frac{\partial E}{\partial \dot{x}^i} - \frac{\partial E}{\partial x^i} = 0,
\Rightarrow \dot{x}^i \frac{\partial^2 E}{\partial \dot{x}^i \partial \dot{x}^j} + x^i \frac{\partial^2 E}{\partial \dot{x}^i \partial \dot{x}^j} \frac{\partial E}{\partial x^j} = 0
\Rightarrow \ddot{x}^i = f^i(x, \dot{x}),
\]

The inverse problem: metrizability

Metric \( \iff \) SODE

Remark: metrizability \( \iff \) Paolo: dynamical potential?
Calculus of variations:
- scalar product $\langle v, v \rangle = E(v)$  
  Riemann: $\langle v, v \rangle_x$  
  Finsler: $\langle v, v \rangle_{x,y}$
- $I[\gamma] = \int_\gamma E(\gamma, \dot{\gamma})$
- $\frac{d}{dt} \frac{\partial E}{\partial \dot{x}^i} - \frac{\partial E}{\partial x^i} = 0$,  
  $\Rightarrow \ddot{x}^i - \frac{\partial^2 E}{\partial x^i \partial \dot{x}^j} + \dot{x}^i \frac{\partial^2 E}{\partial \dot{x}^i \partial \dot{x}^j} - \frac{\partial E}{\partial x^i} = 0$  
  $\Rightarrow \ddot{x}^i = f^i(x, \dot{x})$,

Metric $\Rightarrow$ SODE

The inverse problem: metrizability

Metric $\Leftarrow$ SODE

Remark: metrizability $\iff$ Paolo: dynamical potential?

Geometric tools associated to a SODE:
- SODE $\Longrightarrow$ Spray: vectorfield on $TM$  
  $S = (x^i, y^i, \dot{x}^i, f^i(x, y))$
- Path of the spray: $\gamma$  
  $S_{\gamma} = \dot{\gamma}$  
  $\iff \frac{d^2 x^i}{dt^2} = f^i \left( x, \frac{dx}{dt} \right)$.
- Parallel translation...

The Euler-Lagrange PDE system

$\ddot{x}^i = f^i(x, \dot{x})$  
$\Rightarrow \begin{cases} 
\omega_E = y^i \frac{\partial^2 E}{\partial x^i \partial y^j} + f^i \frac{\partial^2 E}{\partial y^i \partial y^j} - \frac{\partial E}{\partial x^j} = 0, \\
y^i \frac{\partial E}{\partial y^i} - 2E = 0,
\end{cases}$

Euler-Lagrange functions

$\mathcal{E}_S = \{ E \mid \omega_E = 0 \}, \quad \mathcal{E}_{S,2} = \{ E \mid \omega_E = 0, \quad \mathcal{L}_C E = 2E \}$.
Parallel translation: geometric construction

- $R \equiv 0$

- $R \not\equiv 0$
Holonomy distribution

\[ \mathcal{H} := \langle HTM \rangle_{Lie} \]

\[ \mathcal{H} = HTM \oplus v\mathcal{H}, \quad \text{Im } R \subset v\mathcal{H}, \]

\[ \mathcal{C}^{\infty}_{\text{hol}} = \{ E \mid \mathcal{L}_X E = 0, \; X \in \mathcal{H} \}, \quad \mathcal{C}^{\infty}_{\text{hol},2} = \{ E \in \mathcal{C}^{\infty}_{\text{hol}} \mid \mathcal{L}_C E = 2E \} , \]

**Proposition:** \( \mathcal{E}_{S,2} = \mathcal{C}^{\infty}_{\text{hol},2} \)

**Property:** \( \mathcal{E}_S, C^{\infty}_{\text{hol}}, \mathcal{E}_{S,2}(=\mathcal{C}^{\infty}_{\text{hol},2}) \) are vector spaces over \( \mathbb{R} \).

**Proposition:** A 1-homogeneous functional combination of 2-homogeneous Euler-Lagrange functions is a 2-homogeneous Euler-Lagrange function.

\[ E(x,y) := \varphi(E_1(x,y), \ldots, E_r(x,y)) \]

\[ \mathcal{L}_X E = \frac{\partial \varphi}{\partial z^1} \cdot \mathcal{L}_X E_1 + \cdots + \frac{\partial \varphi}{\partial z^r} \cdot \mathcal{L}_X E_r = 0, \quad \forall X \in \mathcal{H} \]

**Definition:** The degree of metric freedom \( m_S \) of a metrizable spray \( S \) is the maximal number of functionally independent elements of \( \mathcal{E}_{S,2} \). If the spray \( S \) is non-metrizable, then we set \( m_S = 0 \).

**Theorem:** If \( S \) is metrizable and \( \mathcal{H} \) is regular, then

\[ m_S = \text{codim } \mathcal{H} . \]

**Remark:** From the hypothesis of the Theorem one cannot omit the metrizability. There are examples for not metrizable sprays with \( \text{codim } \mathcal{H} > 0 \).

**Corollary:** If \( S \) is isotropic, then

- \( m_S = 0 \) if and only if \( R \neq 0 \) and \( S \) is not metrizable;
- \( m_S = 1 \) if and only if \( R \neq 0 \) and \( S \) is metrizable;
- \( m_S = n \) if and only if \( R = 0 \).
Explicit examples

- $\text{codim } \mathcal{H} = 0, m_S = 0$: 
  \[ f^i = \sqrt{x^2(y^1)^2 + (y^2)^2 y^i + (-1)^i y^1 y^i}, \]

- $\text{codim } \mathcal{H} = 1, m_S = 1$: 
  \[ f^i = \frac{\mu \langle x, y \rangle}{1 + \mu |x|^2} y^i, \quad \mu \in \mathbb{R} \setminus \{0\}, \]

- $\text{codim } \mathcal{H} = n, m_S = n$: 
  \[ f^i = \frac{\langle a, y \rangle}{1 + \langle a, x \rangle} y^i, \]

- $\text{codim } \mathcal{H} > 0, m_S = 0$: 
  \[ f^1 = \frac{(y^1)^2}{2x^2}, \quad f^2 = 0. \]

References


The Maximal Defect k-Subgroups of Semigroups of Graphs and Digraphs by Karoly Podoski

Károly Podoski

Spectrum of Graph

Algebraic graph theory

- adjacency matrix $A = A(\Gamma)$ of $\Gamma = (V, E)$ is a matrix of 0’s and 1’s with $A_{x,y} = 1$ iff $(x, y) \in E$.
- The multiset of eigenvalues of $A$ is called the spectrum of $A$.

Theorem

The spectrum of graphs and digraphs is invariant for isomorphic (di)graphs.
Graphs and Digraphs

- **Digraph**: $\Gamma = (V, E)$, with vertices $V$, edges $E \subseteq V \times V$.
- The reverse edge to $e = (x, y) \in E$ is $\bar{e} = (y, x)$.
- *(Undirected)* Graph: $E$ is symmetric.
- No self-loops: $(x, x) \notin E$.

Spectrum of Graph

**Algebraic graph theory**

- *adjacency matrix* $A = A(\Gamma)$ of $\Gamma = (V, E)$ is a matrix of 0’s and 1’s with $A_{x,y} = 1$ iff $(x, y) \in E$.
- The multiset of eigenvalues of $A$ is called the *spectrum* of $A$.

**Theorem**

The spectrum of graphs and digraphs is invariant for isomorphic (di)graphs.

**Example (Co-spectral Graphs)**

$K_{1,4}$ and $C_4 \sqcup K_1$ have same spectrum.
Spectrum of Graph

Algebraic graph theory

- Adjacency matrix $A = A(\Gamma)$ of $\Gamma = (V, E)$ is a matrix of 0's and 1's with $A_{x,y} = 1$ iff $(x, y) \in E$.
- The multiset of eigenvalues of $A$ is called the spectrum of $A$.

Theorem

The spectrum of graphs and digraphs is invariant for isomorphic (di)graphs.

Example (Co-spectral Graphs)

$K_{1,4}$ and $C_4 \sqcup K_1$ have same spectrum. \(\implies\) not complete invariant

Elementary Collapsings of Directed Edges

Digraph $\Gamma = (V, E)$

Elementary collapsing

edge $e = (x, y)$ $\longrightarrow$ function $T_{x,y} : V \rightarrow V$

$$T_{x,y}(v) = \begin{cases} y & \text{if } v = x, \\ v & \text{otherwise.} \end{cases}$$

Example
Elementary Collapsings of Directed Edges

Digraph $\Gamma = (V, E)$

Elementary collapsing
edge $e = (x, y) \rightarrow$ function $T_{x,y}: V \rightarrow V$

$$T_{x,y}(v) = \begin{cases} y & \text{if } v = x, \\ v & \text{otherwise.} \end{cases}$$

Example

$$T_{(1,2)}: 1 \rightarrow 2$$
$$2 \rightarrow 2$$
$$3 \rightarrow 3$$
$$4 \rightarrow 4$$

Motivation: Biochemical Reactions

Biochemical transitions are modelled as products of commuting elementary collapsings, $f = \prod T_{a,b}$, where $T_{x,y}$ and $T_{y,z}$ do not both occur among the $T_{a,b}$ for any $x$, $y$, and $z$. 
Defect $k$ mappings

Definition
We say a mapping $f : V \to V$ has defect $k$ if $|f(V)| = |V| - k$.

- Permutations have defect 0.
- $T_{x,y}$ has defect 1. That is, its image has cardinality $|V| - 1$.
- $T_{x,y}^2 = T_{x,y}$ is an idempotent.
Defect $k$ mappings

**Definition**
We say a mapping $f : V \rightarrow V$ has defect $k$ if $|f(V)| = |V| - k$.

- Permutations have defect 0.
- $T_{x,y}$ has defect 1. That is, its image has cardinality $|V| - 1$.
- $T_{x,y}^2 = T_{x,y}$ is an idempotent.

$f$ has defect $k \implies fg$ and $gf$ have defect at least $k$.

Composition of Elementary Collapsings

Example ($\Gamma$ is the 3-cycle with edges $(1, 2), (2, 3), (3, 1)$)
compose functions from left to right
Composition of Elementary Collapsings

Example ($\Gamma$ is the 3-cycle with edges (1, 2), (2, 3), (3, 1))

compose functions from left to right

$$T_{1,2}T_{2,3} : 1 \rightarrow 2 \rightarrow 3$$
$$2 \rightarrow 2 \rightarrow 3$$
$$3 \rightarrow 3 \rightarrow 3$$

$$T_{2,3}T_{1,2} : 1 \rightarrow 1 \rightarrow 2$$
$$2 \rightarrow 3 \rightarrow 3$$
$$3 \rightarrow 3 \rightarrow 3$$
Composition of Elementary Collapsings

Definition
The transformation semigroup of a digraph \( \Gamma = (V, E) \) to be the collection of transformations of \( V \) generated by the \( T_e \) (\( e \in E \)).

\[
S(\Gamma) = \langle T_{x,y} \in V^V \mid (x, y) \text{ is an edge of } \Gamma \rangle.
\]

This is also called the semigroup of flows on \( \Gamma \).

Example (\( \Gamma \) is the 3-cycle with edges \((1, 2), (2, 3), (3, 1)\))

compose functions from left to right

\[
T_{1,2}T_{2,3} : 1 \to 2 \to 3 \quad T_{2,3}T_{1,2} : 1 \to 1 \to 2
\]
\[
2 \to 2 \to 3 \quad 2 \to 3 \to 3
\]
\[
3 \to 3 \to 3 \quad 3 \to 3 \to 3
\]
Semigroup of Flows on a Graph

Theorem
$S(\Gamma)$ is an invariant for graphs and for digraphs.

Proof.
Isomorphic (di)graphs obviously have isomorphic semigroups of flows.

Theorem (Nehaniv–Rhodes)
$S(\Gamma)$ is a complete algebraic invariant for undirected graphs.
Semigroup of Flows on a Graph

Example
There exist non-isomorphic digraphs with the same flow semigroup. 3-cyclic and 3-cycle with one reverse edge.

\[ f = T_{2,3} T_{1,2} T_{3,1}: 1 \rightarrow 1 \rightarrow 2 \rightarrow 2 \]
\[ 2 \rightarrow 3 \rightarrow 3 \rightarrow 1 \]
\[ 3 \rightarrow 3 \rightarrow 3 \rightarrow 1 \]
Semigroup of Flows on a Graph

Example
There exist non-isomorphic digraphs with the same flow semigroup. 3-cyclic and 3-cycle with one reverse edge.

\[ f = T_{2,3} T_{1,2} T_{3,1} : 1 \to 1 \to 2 \to 2 \]
\[ \quad 2 \to 3 \to 3 \to 1 \]
\[ \quad 3 \to 3 \to 3 \to 1 \]

Now \( f^2 = T_{3,2} \), corresponding to an edge \((3, 2)\).

Complete Invariance for Graphs

Lemma
If \( T_{x,y} = T_{x_1,y_1} \cdots T_{x_k,y_k} \), then \( T_{x,y} \) or \( T_{y,x} \) appears among \( T_{x_i,y_i} \).
Lemma
If $T_{x,y} = T_{x_1,y_1} \cdots T_{x_k,y_k}$, then $T_{x,y}$ or $T_{y,x}$ appears among $T_{x_i,y_i}$.

Corollary
Let $\Gamma$ be a digraph.
If $T_{x,y} \in S(\Gamma)$ then either $e = (x, y)$ or $\bar{e} = (y, x)$ is an edge of $\Gamma$.

Complete Invariance for Graphs

Lemma
If $T_{x,y} = T_{x_1,y_1} \cdots T_{x_k,y_k}$, then $T_{x,y}$ or $T_{y,x}$ appears among $T_{x_i,y_i}$.

Corollary
Let $\Gamma$ be a digraph.
If $T_{x,y} \in S(\Gamma)$ then either $e = (x, y)$ or $\bar{e} = (y, x)$ is an edge of $\Gamma$.

$\implies$ For graph $\Gamma$, we can recover edges from flow semigroup.
Complete Invariance for Graphs

Lemma
If $T_{x,y} = T_{x_1,y_1} \cdots T_{x_k,y_k}$, then $T_{x,y}$ or $T_{y,x}$ appears among $T_{x_1,y_1}$. 

Corollary
Let $\Gamma$ be a digraph.
If $T_{x,y} \in S(\Gamma)$ then either $e = (x, y)$ or $\bar{e} = (y, x)$ is an edge of $\Gamma$.

$\implies$ For graph $\Gamma$, we can recover edges from flow semigroup.
$\implies$ $S(\Gamma)$ is a complete invariant for graphs.

Reversing Edges in Directed Cycles

$\Delta$ directed $n$-cycle: $n$ nodes, edges $(1, 2), \ldots, (n-1, n), (n, 1)$. 
Δ directed $n$-cycle: $n$ nodes, edges $(1, 2), \ldots, (n - 1, n), (n, 1)$.

$W = T_{n,1} T_{n-1,n} T_{n-2,n-1} \cdots T_{1,2}$

$W: 1 \rightarrow 2$

$2 \rightarrow 3$

$\vdots$

$i \rightarrow i + 1$

$\vdots$

$n - 1 \rightarrow n$

$n \rightarrow 2$
Reversing Edges in Directed Cycles

$\Delta$ directed $n$-cycle: $n$ nodes, edges $(1, 2), \ldots, (n-1, n), (n, 1)$.  
$W = T_{n,1} T_{n-1,n} T_{n-2,n-1} \cdots T_{1,2}$

$W: 1 \rightarrow 2$  
$2 \rightarrow 3$  
\vdots  
$i \rightarrow i + 1$  
\vdots  
$n-1 \rightarrow n$  
$n \rightarrow 2$

Thus, $W = (2, 3, \ldots, n-1, n) T_{1,2}$

$W: 1 \rightarrow 2$  
$2 \rightarrow 3$  
\vdots  
$i \rightarrow i + 1$  
\vdots  
$n-1 \rightarrow n$  
$n \rightarrow 2$

$W^{n-1}$ is identity on $V \setminus \{1\}$, sends 1 to $n$
Reversing Edges in Directed Cycles

$\Delta$ directed $n$-cycle: $n$ nodes, edges $(1, 2), \ldots, (n-1, n), (n, 1)$.

$W = T_{n,1} T_{n-1,n} T_{n-2,n-1} \ldots T_{1,2}$

$W: 1 \rightarrow 2$  Thus, $W = (2, 3, \ldots, n-1, n) T_{1,2}$

$2 \rightarrow 3$  $W^{n-1}$ is identity on $V \setminus \{1\}$, sends 1 to $n$

$\vdots$

That is, $W^{(n-1)} = T_{1,n} \in S(\Delta)$.  

$i \rightarrow i + 1$

$\vdots$

$n-1 \rightarrow n$

$n \rightarrow 2$

Reversing Edges in Directed Cycles

$\Delta$ directed $n$-cycle: $n$ nodes, edges $(1, 2), \ldots, (n-1, n), (n, 1)$.

$W = T_{n,1} T_{n-1,n} T_{n-2,n-1} \ldots T_{1,2}$

$W: 1 \rightarrow 2$  Thus, $W = (2, 3, \ldots, n-1, n) T_{1,2}$

$2 \rightarrow 3$  $W^{n-1}$ is identity on $V \setminus \{1\}$, sends 1 to $n$

$\vdots$

That is, $W^{(n-1)} = T_{1,n} \in S(\Delta)$.  

$i \rightarrow i + 1$

$\vdots$

$n-1 \rightarrow n$

$n \rightarrow 2$

So, $S(\Delta \cup \{(1, n)\}) = S(\Delta)$. 

Reversing Edges in Directed Cycles
Reversing Edges in Directed Cycles

Theorem
Digraph $\Gamma$. If $e$ is an edge in a simple directed cycle in $\Gamma$, then $S(\Gamma) = S(\Gamma \cup \{\bar{e}\})$. $\implies$ Enough to consider undirected graphs on strongly connected components.

$\Delta$ directed $n$-cycle: $n$ nodes, edges $(1, 2), \ldots, (n - 1, n), (n, 1)$.
$W = T_{n, 1} T_{n-1, n} T_{n-2, n-1} \ldots T_{1, 2}$
$W: 1 \to 2$ Thus, $W = (2, 3, \ldots n - 1, n) T_{1, 2}$
$2 \to 3$ $W^{n-1}$ is identity on $V \setminus \{1\}$, sends 1 to $n$
$\vdots$ That is, $W^{(n-1)} = T_{1, n} \in S(\Delta)$.
$i \to i + 1$
$\vdots$
$\vdots$
$n - 1 \to n$
$n \to 2$

Defect $k$ Permutator Groups

Permutator Group

$\blacktriangleright$ If $e^2 = e: V \to V$ is an idempotent $S(\Gamma)$.
Let $X_e = V \cdot e = \{v \cdot e | v \in V\}$
Let $G_e = \text{(unique) maximal subgroup of } S(\Gamma) \text{ containing } e$.

$\blacktriangleright$ $(X_e, G_e)$ (faithful) permutator group of subset $X_e$, consists of defect $k$ where $k = |V| - |X_e|$.

Theorem
The defect $k$ permutator groups (up to isomorphism) are invariants for digraphs.
Reversing Edges in Directed Cycles

**Theorem**

Digraph $\Gamma$. If $e$ is an edge in a simple directed cycle in $\Gamma$, then $S(\Gamma) = S(\Gamma \cup \{e\})$. $\rightarrow$ Enough to consider undirected graphs on strongly connected components.

$\Delta$ directed $n$-cycle: $n$ nodes, edges $(1, 2), \ldots, (n-1, n), (n, 1)$. $W = T_{n, 1}T_{n-1, n}T_{n-2, n-1} \cdots T_{1, 2}$

$W: 1 \rightarrow 2 \quad \text{Thus, } W = (2, 3, \ldots, n-1, n)T_{1, 2}$

$2 \rightarrow 3 \quad W^{n-1}$ is identity on $V \setminus \{1\}$, sends $1$ to $n$

$\vdots \quad \text{That is, } W^{(n-1)} = T_{1, n} \in S(\Delta)$.

$i \rightarrow i+1 \quad W$ defect $1$: permutes $n-1$ vertices cyclically

$\vdots \quad W^{n-1}$

$n-1 \rightarrow n$

$n \rightarrow 2$

**Defect $k$ Permutator Groups**

**Permutator Group**

- If $e^2 = e: V \rightarrow V$ is an idempotent $S(\Gamma)$.
  Let $X_e = V \cdot e = \{v \cdot e \mid v \in V\}$
  Let $G_e$ = (unique) maximal subgroup of $S(\Gamma)$ containing $e$.

- $(X_e, G_e)$ (faithful) permutator group of subset $X_e$, consists of defect $k$ where $k = |V| - |X_e|$.

**Theorem**
The defect $k$ permutator groups (up to isomorphism) are invariants for digraphs.
Some Defect 1 Permutator Groups

Example
- Cycle graph with $n$ nodes: Defect 1 group: cyclic $C_{n-1}$

Example
- Cycle graph with $n$ nodes: Defect 1 group: cyclic $C_{n-1}$
- Two cycle graphs $n$ and $m$ nodes:
Some Defect 1 Permutator Groups

Example

- Cycle graph with $n$ nodes: Defect 1 group: cyclic $C_{n-1}$
- Two cycle graphs $n$ and $m$ nodes: Defect 1 group:
  $C_{n-1} \times C_{m-1}$

Example

- Cycle graph with $n$ nodes: Defect 1 group: cyclic $C_{n-1}$
- Two cycle graphs $n$ and $m$ nodes: Defect 1 group:
  $C_{n-1} \times C_{m-1}$
- Overlapping Cycles:
Some Defect 1 Permutator Groups

Example

- Cycle graph with $n$ nodes: Defect 1 group: cyclic $C_{n-1}$
- Two cycle graphs $n$ and $m$ nodes: Defect 1 group: $C_{n-1} \times C_{m-1}$
- Overlapping Cycles: Defect 1 group: $S_{n-1}$ or $A_{n-1}$ or $S_5$

Defect 1 Permutator Groups

Theorem

$$T_{k,l,m} = \langle x, y \rangle, \quad n = k + l + m,$$
$$x = (a_1, a_2, \ldots, a_k, b_1, b_2, \ldots, b_l),$$
$$y = (a_1, a_2, \ldots, a_k, c_1, c_2, \ldots, c_m),$$

1. If $k = 0$ then $T_{0,l,m} \cong C_l \times C_m$;
2. If $k \geq 1$, $k + l$ and $k + m$ are both odd, then $T_{k,l,m} = A_n$;
3. $T_{3,2,1} \cong T_{2,2,2} \cong T_{3,1,2} \cong S_5$, and this is a 3-transitive action of $S_5$ on 6 elements;
4. $T_{k,l,m} = S_n$, otherwise.
Defect 1 Permutator Groups

- ear decomposition

Defect 1 Permutator Groups

- ear decomposition $\iff$ 2-edge connected graph
Defect 1 Permutator Groups

- ear decomposition $\iff$ 2-edge connected graph
- $k$-edge connected graph: removing any $k - 1$ edges does not disconnect the graph

- open ear decomposition
Defect 1 Permutator Groups

- ear decomposition \iff 2-edge connected graph
- \( k \)-edge connected graph: removing any \( k - 1 \) edges does not disconnect the graph
- open ear decomposition \iff overlap in at least 2 vertices

- ear decomposition \iff 2-edge connected graph
- \( k \)-edge connected graph: removing any \( k - 1 \) edges does not disconnect the graph
- open ear decomposition \iff overlap in at least 2 vertices
  \iff 2-vertex connected graph
Defect 1 Permutator Groups

- ear decomposition $\iff$ 2-edge connected graph
- $k$-edge connected graph: removing any $k - 1$ edges does not disconnect the graph
- open ear decomposition $\iff$ overlap in at least 2 vertices $\iff$ 2-vertex connected graph
- $k$-vertex connected graph: removing any $k - 1$ vertices does not disconnect the graph

- done for all 2-vertex connected graphs
Defect 1 Permutator Groups

- ear decomposition $\iff$ 2-edge connected graph
- $k$-edge connected graph: removing any $k - 1$ edges does not disconnect the graph
- open ear decomposition $\iff$ overlap in at least 2 vertices $\iff$ 2-vertex connected graph
- $k$-vertex connected graph: removing any $k - 1$ vertices does not disconnect the graph
- done for all 2-vertex connected graphs
- direct products of defect 1 groups of biconnected components

Main Theorem (Horváth, Nehaniv, Podoski, 2015)
Graph $\Gamma$, 2-edge connected, not a cycle.
Then maximal defect $k$ groups are all isomorphic and
- Defect 1: Direct product of cyclic, alternating and symmetric groups of various orders, (corresponding to the biconnected components)
- Defect $k \geq 2$, no bridge with $\geq k$ nodes $\implies S_{n-k}$
- Defect $k \geq 2$, bridges with $\geq k$ nodes $\implies$ direct product of symmetric groups (corresponding to the biconnected components with $\leq k - 1$ long bridges)
Acknowledgment

This work was supported in part by the European Commission FP7 project BIOMICS, contract number CNECT-318202 to University of Hertfordshire (UK), University of Debrecen (HU), Passau University (DE), and University of Dundee (UK).
12 How can we find ordinary differential equations having a given a symmetry group by Ágota Figula

How can we find ordinary differential equations having a given symmetry group?

Ágota Figula

University of Debrecen

Passau, 8-10.02.2016

Preliminaries

Symmetries of a differential equation are transformations that move continuously a solution of the equation into another solution. Thus for each symmetry there exists a corresponding vector field (the *infinitesimal generator* of the symmetry). In the case of ordinary differential equations of order \( m \) the space of the variables \( x, y, y', \ldots, y^{(m)} \) is called the jet space. The differential equation \( f(x, y, y', \ldots, y^{(m)}) = 0 \) defines an \( m + 1 \)-dimensional surface in this space which is called the hull of the differential equation. A smooth solution is a continuously differentiable function \( \varphi(x) \) such that the curve \( y = \varphi(x) \) with \( y' = \frac{d\varphi(x)}{dx}, \ldots, y^{(m)} = \frac{d^m\varphi(x)}{dx^m} \) belongs to the hull, that is, \( f(x, \varphi(x), \ldots, \frac{d^m\varphi(x)}{dx^m}) = 0 \) identically for all \( x \) holds.
Preliminaries
The symmetries of a differential equation form a group, which is called the symmetry group. It is the group of transformations of the \((x, y)\)-plane the prolongation of which to the derivatives \(y', \ldots, y^{(m)}\) leaves the hull of the equation under consideration invariant. The symmetries of a differential equation one can use for generating from a known solution of the differential equation a new solution and creating new methods for solving it, for example lowering the order of the differential equation. In the process of integrating differential equations the important step is the simplification of the hull by a suitable change of variables.

Lie’s method
Lie has determined the groups of transformations of the \((x, y)\)-plane and written these into canonical form. He provided a classification of all ordinary differential equations of arbitrary order which admit these given groups as groups of their symmetries. We wish to present his method and give examples.

Let \( G \) be a given \( r \)-dimensional real Lie group of transformation of the \((x, y)\)-plane. It is given by the basis elements (the infinitesimal generators) of its tangential Lie algebra \( g \):

\[
\begin{align*}
X_1 &= \phi_1(x, y) \frac{\partial}{\partial x} + \eta_1(x, y) \frac{\partial}{\partial y}, \\
X_2 &= \phi_2(x, y) \frac{\partial}{\partial x} + \eta_2(x, y) \frac{\partial}{\partial y}, \\
&\vdots \\
X_r &= \phi_r(x, y) \frac{\partial}{\partial x} + \eta_r(x, y) \frac{\partial}{\partial y}.
\end{align*}
\]
The $m$-th prolonged vector fields $X_i^{(m)}$, $i = 1, 2, \ldots, r$, are defined as

$$X_i^{(m)} = \phi_i(x, y) \frac{\partial}{\partial x} + \eta_i(x, y) \frac{\partial}{\partial y} + \eta_i^{(1)}(x, y, y') \frac{\partial}{\partial y'} + \cdots + \eta_i^{(m)}(x, y, \ldots, y^{(m)}) \frac{\partial}{\partial y^{(m)}},$$

(2)

where $\eta_i^{(k)}$, $k = 1, 2, \ldots, m$, is the $k$-th coordinate function of the prolongation of the vector field $X_i$. We define recursively the functions $\eta_i^{(k)}$ as follows:

$$\eta_i^{(k)} = \frac{d\eta_i^{(k-1)}}{dx} - y^{(k)} \frac{d\phi_i}{dx},$$

that is

$$\eta_i^{(1)}(x, y, y') = \frac{\partial \eta_i(x, y)}{\partial x} + \frac{\partial \eta_i(x, y)}{\partial y} y' - \frac{\partial \phi_i(x, y)}{\partial x} y' - \frac{\partial \phi_i(x, y)}{\partial y} (y')^2,$$

$$\eta_i^{(2)}(x, y, y', y^{(2)}) = \frac{\partial \eta_i^{(1)}(x, y, y')}{\partial x} + \frac{\partial \eta_i^{(1)}(x, y, y')}{\partial y} y' +$$

$$+ \frac{\partial \eta_i^{(1)}(x, y, y')}{\partial y'} y^{(2)} - \frac{\partial \phi_i(x, y)}{\partial x} y^{(2)} - \frac{\partial \phi_i(x, y)}{\partial y} y' y^{(2)},$$

(3)

and so further.

The $m$-th prolonged vector fields $X_i^{(m)}$, $i = 1, 2, \ldots, r$, depend on $x, y, y', \ldots, y^{(m)}$ and they are generators of the Lie algebra $\mathfrak{g}$. 
A differential equation

\[ f(x, y, y', \ldots, y^{(m)}) = 0 \]

of order \( m \) admits a group of symmetries whose Lie algebra is \( g \) if and only if the following system of partial differential equations is satisfied:

\[
\begin{align*}
\phi_1 \frac{\partial f}{\partial x} + \eta_1 \frac{\partial f}{\partial y} + \eta_1^{(1)} \frac{\partial f}{\partial y^1} + \cdots + \eta_1^{(m)} \frac{\partial f}{\partial y^{(m)}} &= 0 \\
\phi_2 \frac{\partial f}{\partial x} + \eta_2 \frac{\partial f}{\partial y} + \eta_2^{(1)} \frac{\partial f}{\partial y^1} + \cdots + \eta_2^{(m)} \frac{\partial f}{\partial y^{(m)}} &= 0 \\
& \cdots \\
\phi_r \frac{\partial f}{\partial x} + \eta_r \frac{\partial f}{\partial y} + \eta_r^{(1)} \frac{\partial f}{\partial y^1} + \cdots + \eta_r^{(m)} \frac{\partial f}{\partial y^{(m)}} &= 0
\end{align*}
\]

whenever \( f(x, y, y', \ldots, y^{(m)}) = 0 \) holds.

Let \( M \) be the matrix

\[
M = \begin{pmatrix}
\phi_1 & \phi_2 & \phi_3 & \cdots & \phi_r \\
\eta_1 & \eta_2 & \eta_3 & \cdots & \eta_r \\
\eta_1^{(1)} & \eta_2^{(1)} & \eta_3^{(1)} & \cdots & \eta_r^{(1)} \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
\eta_1^{(m)} & \eta_2^{(m)} & \eta_3^{(m)} & \cdots & \eta_r^{(m)}
\end{pmatrix}.
\]

Then the system of partial differential equations given by (5) can be treated as the following system of linear equations in the variables \( \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \cdots, \frac{\partial f}{\partial y^{(m)}} \):

\[
\begin{pmatrix}
\frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} & \frac{\partial f}{\partial y^1} & \cdots & \frac{\partial f}{\partial y^{(m)}}
\end{pmatrix} \cdot M = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \end{pmatrix}.
\]
The case $m = r - 2$

The coefficient matrix $M$ is an $(m + 2) \times r$ matrix. Thus the system (5) has more than only the trivial solution
\[
\frac{\partial f}{\partial x} = \frac{\partial f}{\partial y} = ... = \frac{\partial f}{\partial y^{(m)}} = 0
\]
if and only if the rank of $M$ is strictly less than $(m + 2)$: $rk(M) < m + 2$. Now, $rk(M) \leq r$ always holds. Hence if $m > r - 2$, then the rank condition is automatically satisfied.

First we consider the case $m = r - 2$. In that case one can arrive at the possible differential equations in the following way: Then the matrix $M$ is an $(m + 2) \times (m + 2)$-matrix. The system (5) has a non-trivial solution $f$ if and only if the rank of $M$ is $< m + 2$.

Hence the determinant $D = |M|$ of $M$, which is a polynomial function of $x$, $y$, $y^{(i)}$, $i = 1, 2, ..., m = r - 2$, has to be 0. When $D$ is not identically 0 as a polynomial, then $D$ is a polynomial function of $x$, $y$, $y^{(i)}$, $i = 1, 2, ..., m = r - 2$, which has to be 0 if a non-trivial differential equation $f$ exists. Hence by factoring $D$ we obtain the only possibilities for such $f$.

The case $m < r - 2$

Now, assume $m < r - 2$. Then the coefficient matrix $M$ is an $(m + 2) \times r$-matrix. To obtain a non-trivial solution of the system (5) it is necessary that $rk(M) < m + 2$. Hence the determinants of all $(m + 2) \times (m + 2)$ submatrices of $M$ has to be 0. Again, these determinants are polynomials of the variables $x$, $y$, $y^{(i)}$, $i = 1, 2, ..., m < r - 2$, and therefore their common factors provide the only possibilities for non-trivial differential equations $f$ admitting $G$. 

---

**Agota Figula**

Differential equations having a given symmetry group
Theorem

Finding the differential equations \( f(x, y, y', ..., y^{(m)}) = 0 \) of order \( m \), which admit a group of symmetries whose Lie algebra is a given \( r \)-dimensional real Lie algebra \( g \) such that \( m \leq r - 2 \), one has to build the matrix

\[
M = \begin{pmatrix}
\phi_1 & \phi_2 & \phi_3 & \cdots & \phi_r \\
\eta_1^{(1)} & \eta_2^{(1)} & \eta_3^{(1)} & \cdots & \eta_r^{(1)} \\
\eta_1^{(m)} & \eta_2^{(m)} & \eta_3^{(m)} & \cdots & \eta_r^{(m)}
\end{pmatrix}
\]

and compute the greatest common divisor of all \((m + 2) \times (m + 2)\) subdeterminants. The factors of this polynomial give the only possibilities for the sought differential equations, unless this polynomial is identically 0.


In this list there are three different imprimitive actions of \( sl_2(\mathbb{R}) \) and one primitive action of \( sl_2(\mathbb{R}) \) on the plane. An action is called imprimitive of there exists a foliation of the manifold with lower dimensional manifolds which are invariant under the action.
Example 1:
The Lie algebra $g_1 = \mathfrak{sl}_2(\mathbb{R})$ is generated by the vector fields:

\begin{align*}
X_1 &= \frac{\partial}{\partial x} + \frac{\partial}{\partial y}, \\
X_2 &= x \frac{\partial}{\partial x} + y \frac{\partial}{\partial y}, \\
X_3 &= x^2 \frac{\partial}{\partial x} + y^2 \frac{\partial}{\partial y}.
\end{align*}

By Theorem (1) to obtain all first order ordinary differential equations which are invariant under the symmetry group corresponding to $g_1$ we have to determine the first prolonged vector fields $X^{(1)}_i$. Since one has

\begin{align*}
(\phi_1, \phi_2, \phi_3) &= (1, x, x^2), \\
(\eta_1, \eta_2, \eta_3) &= (1, y, y^2)
\end{align*}

using formula (3) we obtain

\begin{align*}
(\eta^{(1)}_1, \eta^{(1)}_2, \eta^{(1)}_3) &= (0, 0, (2y - 2x)y').
\end{align*}

Since the determinant $D = \begin{vmatrix}
\phi_i \\
\eta_i \\
\eta^{(1)}_i
\end{vmatrix} = \begin{vmatrix}
1 & x & x^2 \\
1 & y & y^2 \\
0 & 0 & (2y - 2x)y'
\end{vmatrix}$

equals $2(y - x)^2y'$ one can see that the first order differential equation $y' = 0$ is invariant under the symmetry group corresponding to the Lie algebra $g_1$ given by (9) (cf. Theorem 1).
The case $m > r - 2$

How we can find the ordinary differential equations of order $m$ which allow a given Lie group of dimension $r$ as the group $G$ of their symmetries such that $m > r - 2$? As before, $G$ is an $r$-dimensional Lie group of transformations acting on $(x, y)$-plane such that the Lie algebra $\mathfrak{g}$ of $G$ is given by the infinitesimal generators: $\mathbf{X}_i$, $i = 1, 2, ..., r$ in (1). We consider the $m$-th prolongations $\mathbf{X}_i^{(m)}$ of the vector field $\mathbf{X}_i$, $i = 1, 2, ..., r$:

$$
\mathbf{X}_i^{(m)} = \phi_i(x, y) \frac{\partial}{\partial x} + \eta_i(x, y) \frac{\partial}{\partial y} + \eta_i^{(1)}(x, y, y') \frac{\partial}{\partial y'} + \cdots + \eta_i^{(m)}(x, y, \ldots, y^{(m)}) \frac{\partial}{\partial y^{(m)}}.
$$

(12)

They act on the $(m + 2)$-dimensional manifold $\{x, y, y', \ldots, y^{(m)}\}$.

Assume that the determinant $D$ in Theorem (1) is not identically zero. Then the system of the partial differential equations given by

$$
\begin{align*}
\phi_1 \frac{\partial f}{\partial x} + \eta_1 \frac{\partial f}{\partial y} + \eta_1^{(1)} \frac{\partial f}{\partial y'} + \cdots + \eta_1^{(m)} \frac{\partial f}{\partial y^{(m)}} &= 0 \\
\phi_2 \frac{\partial f}{\partial x} + \eta_2 \frac{\partial f}{\partial y} + \eta_2^{(1)} \frac{\partial f}{\partial y'} + \cdots + \eta_2^{(m)} \frac{\partial f}{\partial y^{(m)}} &= 0 \\
&\vdots \\
\phi_r \frac{\partial f}{\partial x} + \eta_r \frac{\partial f}{\partial y} + \eta_r^{(1)} \frac{\partial f}{\partial y'} + \cdots + \eta_r^{(m)} \frac{\partial f}{\partial y^{(m)}} &= 0
\end{align*}
$$

(13)

has $m + 2 - r$ common solutions.
Example 1 for $m = 2$

To determine all ordinary differential equations of order 2 which are invariant under the 3-dimensional group of symmetries corresponding to the Lie algebra $g_1$ we need to compute the 2-th prolonged vector fields $X_i^{(2)}$, $i = 1, 2, 3$, of (9). Using (10) we obtain $(\eta_1^{(1)}, \eta_2^{(1)}, \eta_3^{(1)}) = (0, 0, (2y - 2x)y')$,

(14) $(\eta_1^{(2)}, \eta_2^{(2)}, \eta_3^{(2)}) = (0, y^{(2)}, -2y' + 2(y')^2 - 4xy^{(2)} + 2yy^{(2)})$,

This yields the following system of partial differential equations:

(15) \[
\begin{align*}
x \frac{\partial f}{\partial x} + y \frac{\partial f}{\partial y} - y^{(2)} \frac{\partial f}{\partial y^{(2)}} &= 0, \\
x^2 \frac{\partial f}{\partial x} + y^2 \frac{\partial f}{\partial y} + 2(y - x)y' \frac{\partial f}{\partial y'} + \\
+ 2(yy^{(2)} - 2xy^{(2)} + (y')^2 - y') \frac{\partial f}{\partial y^{(2)}} &= 0.
\end{align*}
\]

From the first equation of (15) it follows that $f$ depends on $x - y$. Using this the second and third equation reduces to

(16) \[
\begin{align*}
(y - x) \frac{\partial f}{\partial y} - y^{(2)} \frac{\partial f}{\partial y^{(2)}} &= 0, \\
2(y - x)y' \frac{\partial f}{\partial y'} + (2y^2 - 2y' + 3(y - x)y^{(2)}) \frac{\partial f}{\partial y^{(2)}} &= 0.
\end{align*}
\]

By integration of the first equation we obtain that its solutions are generated by:

(17) $y', \alpha = (y - x)y^{(2)}$. 

Taking these as new variables the vector field

\[ Y = 2(y - x)y' \frac{\partial}{\partial y'} + (2y'(y' - 1) + 3(y - x)y^{(2)}) \frac{\partial}{\partial y^{(2)}} \]

takes the form

\[ \bar{Y} = Y(y') \frac{\partial}{\partial y'} + Y(\alpha) \frac{\partial}{\partial \alpha} = 2y' \frac{\partial}{\partial y'} + (3\alpha + 2y'^2 - 2y') \frac{\partial}{\partial \alpha}. \]

The solution of the partial differential equation

\[ 2y' \frac{\partial f}{\partial y'} + (3\alpha + 2y'^2 - 2y') \frac{\partial f}{\partial \alpha} = 0 \]

is

\[ \varphi_1 = \alpha y' - \frac{\alpha}{2} - 2(y'^{\frac{3}{2}} + y' - \frac{1}{2}) = \]

\[ (y - x)y^{(2)}y'^{-\frac{3}{2}} - 2(y'^{\frac{1}{2}} + y'^{-\frac{1}{2}}). \]

---

**The case \( m + 2 = r + 1 \)**

If \( m + 2 = r + 1 \), then there exists a solution \( \varphi_1 \) of (5) which can be found by integration. The solution \( \varphi_1 \) depends solely on the variables \( x, y, y', \ldots, y^{(r-1)} \). It follows from the following: We have to solve the system of the partial differential equations given by

\[ \begin{align*}
\phi_1 \frac{\partial f}{\partial x} + \eta_1 \frac{\partial f}{\partial y} + \eta_1^{(1)} \frac{\partial f}{\partial y'} + \cdots + \eta_1^{(r-1)} \frac{\partial f}{\partial y^{(r-1)}} &= 0 \\
\phi_2 \frac{\partial f}{\partial x} + \eta_2 \frac{\partial f}{\partial y} + \eta_2^{(1)} \frac{\partial f}{\partial y'} + \cdots + \eta_2^{(r-1)} \frac{\partial f}{\partial y^{(r-1)}} &= 0 \\
&\vdots \\
\phi_r \frac{\partial f}{\partial x} + \eta_r \frac{\partial f}{\partial y} + \eta_r^{(1)} \frac{\partial f}{\partial y'} + \cdots + \eta_r^{(r-1)} \frac{\partial f}{\partial y^{(r-1)}} &= 0.
\end{align*} \]
We have to determine the joint invariants of the \( r - 1 \)-th prolonged vector fields \( X_i^{(r-1)} \), \( i = 1, 2, \ldots, r - 1 \). To obtain these we determine by integration the functionally independent invariants \( \alpha_j(x, y, \ldots, y^{(r-1)}) \) of one of the vector fields, say \( X_1^{(r-1)} \). These invariants are the solutions of the linear homogeneous partial differential equation

\[
X_1^{(r-1)}(\alpha) = \phi_1 \frac{\partial \alpha}{\partial x} + \eta_1 \frac{\partial \alpha}{\partial y} + \eta_1^{(1)} \frac{\partial \alpha}{\partial y'} + \cdots + \eta_1^{(r-1)} \frac{\partial \alpha}{\partial y^{(r-1)}} = 0. 
\]

If \( X_1^{(r-1)}(\alpha) \neq 0 \) in a neighbourhood of a chosen point \( x_0 \), then there are \( r \) functionally independent invariants, hence \( r \) functionally independent solutions of the partial differential equation (21) in a neighbourhood of \( x_0 \).

Since any joint invariant \( \varphi \) must in particular be an invariant of \( X_1^{(r-1)} \), we can write \( \varphi \) as some function of the computed invariants \( \alpha_j, j = 1, 2, \ldots, r \), of \( X_1^{(r-1)} \). Using the invariants \( \alpha_j, j = 1, 2, \ldots, r \), of \( X_1^{(r-1)} \) as new variables (coordinates), we express the remaining vector fields \( X_2^{(r-1)}, \ldots, X_r^{(r-1)} \) in these new coordinates. Then we find joint invariants of these new \( r - 1 \) vector fields \( X_2^{(r-1)}(\alpha_j), \ldots, X_r^{(r-1)}(\alpha_j) \). This procedure works inductively leading eventually to the joint invariants of all the vector fields expressed in terms of the joint invariants of the first \( \alpha_j, j = 1, 2, \ldots, r \), of them. This gives the common solution

\[
\varphi_1(\alpha_j) = \varphi_1(x, y, y', \ldots, y^{(r-1)}) \text{ of (20)}. 
\]
If $m + 2 = r + 2$, then there are two solutions of (5). One is $\varphi_1$, the other is $\varphi_2$ which depends on the variables $x, y, y', \ldots, y^{(r)}$. If $m + 2 = r + 3$, then there are three solutions $\varphi_i, \ i = 1, 2, 3$, such that $\varphi_3$ depends on $x, y, y', \ldots, y^{(r+1)}$. For arbitrary $m$ there are $m + 2 - r$ solutions $\varphi_j, \ j = 1, 2, \ldots, m + 2 - r$. Now we show that it is enough to determine the solutions $\varphi_1, \varphi_2$ by integration. If we know the solutions $\varphi_1, \varphi_2$, then for all $i \geq 3$ the solutions $\varphi_i$ can be found using $\varphi_1$ and $\varphi_2$ by differentiation in the following way:

The equation

$$\varphi_2 - a\varphi_1 + b = 0$$

with arbitrary constants $a, b$ is a differential equation of order $r$ which is invariant under the action of $g$.

By differentiation with respect to the variable $x$ one gets that

$$\frac{d\varphi_2}{dx} - a\frac{d\varphi_1}{dx} = 0$$

or equivalently

$$(22) \quad \frac{d\varphi_2}{dx} : \frac{d\varphi_1}{dx} = a.$$  

(22) is a differential equation of order $r + 1$ which is invariant under the action of $g$. Hence one can choose $\frac{d\varphi_2}{dx} : \frac{d\varphi_1}{dx}$ as $\varphi_3$ and further $\frac{d\varphi_3}{dx} : \frac{d\varphi_1}{dx}$ as $\varphi_4$. 
**Theorem**

Any differential equation of order \( m > r + 2 \) which admits the symmetry group the Lie algebra of which is the \( r \)-dimensional real Lie algebra \( \mathfrak{g} \) given by

\[
X_i = \phi_i(x, y) \frac{\partial}{\partial x} + \eta_i(x, y) \frac{\partial}{\partial y}, \quad i = 1, 2, \ldots, r,
\]

has the form

\[
\Omega(\varphi_1, \varphi_2, \varphi_3, \ldots) = 0,
\]

with some smooth function \( \Omega \).

For the concrete calculation of differential equations of order \( m > r + 2 \) which admit the symmetry group the Lie algebra of which is given by the vector fields: \( X_i, \quad i = 1, 2, \ldots, r \), we proceed in the following way:

1. We have to find two common solutions \( \varphi_1, \varphi_2 \) of the system of partial differential equations given by (5) with \( m = r \) such that \( \varphi_1 \) is a function of \( x, y, \ldots, y^{(r-1)} \) and \( \varphi_2 \) is a function of \( x, y, \ldots, y^{(r)} \). We compute the \( r \)-th prolonged vector fields \( X_i^{(r)} \), \( i = 1, 2, \ldots, r \).
2. Then we have to determine the two joint invariants of all the $r$-th prolonged vector fields $X_i^{(r)}$, $i = 1, 2, \ldots, r$. To obtain these we determine by integration the functionally independent invariants $\alpha_j(x, y, \ldots, y^{(r)})$ of one of the vector fields, say $X_i^{(r)}$. These invariants are the solutions of the linear homogeneous partial differential equation

\begin{equation}
X_1^{(r)}(\beta) = \phi_1 \frac{\partial \beta}{\partial x} + \eta_1 \frac{\partial \beta}{\partial y} + \eta_1^{(1)} \frac{\partial \beta}{\partial y^1} + \cdots + \eta_1^{(r)} \frac{\partial \beta}{\partial y^{(r)}} = 0.
\end{equation}

If $X_1(\beta) \neq 0$ in a neighbourhood of a chosen point $x_0$, then there are $r + 1$ functionally independent invariants, hence $r + 1$ functionally independent solutions of the partial differential equation (23) in a neighbourhood of $x_0$.

3. Since any joint invariant $\varphi$ must in particular be an invariant of $X_1^{(r)}$, we can write $\varphi$ as some function of the computed invariants $\beta_j$, $j = 1, 2, \ldots, r + 1$, of $X_1^{(r)}$. Using the invariants $\beta_j$, $j = 1, 2, \ldots, r + 1$, of $X_1^{(r)}$ as new variables (coordinates), we express the remaining vector fields $X_2^{(r)}, \ldots, X_r^{(r)}$ in these new coordinates.

4. Then we find joint invariants of these new $r - 1$ vector fields $X_2^{(r)}(\beta_j), \ldots, X_r^{(r)}(\beta_j)$. This procedure works inductively leading eventually to the joint invariants of all the vector fields expressed in terms of the joint invariants of the first $\beta_j$, $j = 1, 2, \ldots, r + 1$, of them.
Denote by $K$ the subset of the set of indexes $\{1, 2, \ldots, r + 1\}$ such that the invariants $\beta_k$ does not depend on the variable $y^{(r)}$. The invariant $\varphi_1$ is a smooth function of these invariants $\beta_k$, $k \in K$, i.e. $\varphi_1 = \Psi_1(\beta_{k_1}, \beta_{k_2}, \ldots)$, $k_i \in K$. In particular $\varphi_1$ is the joint invariant of all the $r - 1$-th prolonged vector fields $X_i^{(r-1)}$, $i = 1, 2, \ldots, r$.

Therefore one has $\varphi_2 = \Psi_2(\varphi_1, \gamma_1, \gamma_2, \ldots)$, where $l_i \in \{1, 2, \ldots, r + 1\} \setminus K$-s are the indexes of the invariants which depend on $y^{(r)}$.

5. We obtain the further common solutions $\varphi_1$, $i = 3, 4, \ldots$, of the system of partial differential equations (5) with arbitrary $m$, when we use the rule $\varphi_1 = \frac{d\varphi_{1-1}}{dx} : \frac{d\varphi_1}{dx}$.

Example 1:

To determine all ordinary differential equations of order $\geq 2$ which are invariant under the 3-dimensional group of symmetries corresponding to the Lie algebra $g_1$ we need to compute the 3-th prolonged vector fields $X_i^{(3)}$, $i = 1, 2, 3$, of (9). Using (10) we obtain $(\eta_1^{(1)}, \eta_2^{(1)}, \eta_3^{(1)}) = (0, 0, (2y - 2x)y')$,

(24) $(\eta_1^{(2)}, \eta_2^{(2)}, \eta_3^{(2)}) = (0, -y^{(2)}, -2y' + 2(y')^2 - 4xy^{(2)} + 2yy^{(2)})$,

(25) $(\eta_1^{(3)}, \eta_2^{(3)}, \eta_3^{(3)}) = (0, -2y^{(3)}, -6y^{(2)} + 6y^{(2)}y' - 6xy^{(3)} + 2yy^{(3)})$. 

This yields the following system of partial differential equations:

\[
(26) \quad x \frac{\partial f}{\partial x} + y \frac{\partial f}{\partial y} - y(2) \frac{\partial f}{\partial y(2)} - 2y(3) \frac{\partial f}{\partial y(3)} = 0
\]
\[
x^2 \frac{\partial f}{\partial x} + y^2 \frac{\partial f}{\partial y} + 2(y - x)y' \frac{\partial f}{\partial y'} + 2(yy(2) - 2xy(2) + (y')^2 - y') \frac{\partial f}{\partial y(2)} + 2(yy(3) - 3xy(3) - 3y(2) + 3y(2)y') \frac{\partial f}{\partial y(3)} = 0.
\]

From the first equation of (26) it follows that \( f \) depends on \( x - y \).
Using this the second and third equation reduces to

\[
(27) \quad (y - x) \frac{\partial f}{\partial y} - y(2) \frac{\partial f}{\partial y(2)} - 2y(3) \frac{\partial f}{\partial y(3)} = 0
\]
\[
2(y - x)y' \frac{\partial f}{\partial y'} + (2y^2 - 2y' + 3(y - x)y(2)) \frac{\partial f}{\partial y(2)} + (6y(2)y' - 6y(2) + 4(y - x)y(3)) \frac{\partial f}{\partial y(3)} = 0.
\]

By integration of the first equation we obtain that its solutions are generated by:

\[
(28) \quad y', \; \alpha = (y - x)y(2), \; \beta = (y - x)^2y(3).
\]

Taking these as new variables the vector field

\[
Y = 2(y - x)y' \frac{\partial}{\partial y'} + (2y'(y' - 1) + 3(y - x)y(2)) \frac{\partial}{\partial y(2)} + (6y(2)(y' - 1) + 4(y - x)y(3)) \frac{\partial}{\partial y(3)}
\]

takes the form

\[
\tilde{Y} = Y(y') \frac{\partial}{\partial y'} + Y(\alpha) \frac{\partial}{\partial \alpha} + Y(\beta) \frac{\partial}{\partial \beta} =
\]
\[
(29) \quad 2y' \frac{\partial}{\partial y'} + (3\alpha + 2y'^2 - 2y') \frac{\partial}{\partial \alpha} + (4\beta + 6\alpha(y' - 1)) \frac{\partial}{\partial \beta}.
\]
The solutions of the partial differential equation

\[ 2y' \frac{\partial f}{\partial y'} + (3\alpha + 2y'^2 - 2y') \frac{\partial f}{\partial \alpha} + (4\beta + 6\alpha(y' - 1)) \frac{\partial f}{\partial \beta} = 0 \]

are generated by

\[ \varphi_1 = \alpha y'^{-\frac{3}{2}} - 2(y'^{\frac{1}{2}} + y'^{-\frac{1}{2}}) = \]

(30)

\[ (y - x)y^{(2)}y'^{-\frac{3}{2}} - 2(y'^{\frac{1}{2}} + y'^{-\frac{1}{2}}), \]

\[ \varphi_2 = \beta y'^{-2} - 6\varphi_1(y'^{\frac{1}{2}} + y'^{-\frac{1}{2}}) - 6(y' + y'^{-1}) = \]

(31)

\[ (y - x)^2y^{(3)}y'^{-2} - 6(y - x)y^{(2)}(y'^{-2} + y'^{-1}) + 18y' + 24 + 18y'^{-1}. \]

According to Theorem (2) the ordinary differential equations of order \( \geq 2 \) which are invariant under the 3-dimensional group of symmetries corresponding to the Lie algebra \( g_1 \) has the form

\[ \Omega(\varphi_1, \varphi_2, \varphi_3, \cdots) = 0, \]

where \( \varphi_1, \varphi_2 \) has the form (30), (31) and for \( i = 3, 4, \ldots \) one has

\[ \varphi_i = \frac{d\varphi_{i-1}}{dx} : \frac{d\varphi_1}{dx}. \]
The Lie algebra $g_2 = \mathfrak{sl}_2(\mathbb{R})$ is generated by the vector fields:

\[
X_1 = \frac{\partial}{\partial x}, \quad X_2 = 2x \frac{\partial}{\partial x} + y \frac{\partial}{\partial y}, \quad X_3 = x^2 \frac{\partial}{\partial x} + xy \frac{\partial}{\partial y}.
\]

As one has

\[
(\phi_1, \phi_2, \phi_3) = (1, 2x, x^2), \quad (\eta_1, \eta_2, \eta_3) = (0, y, xy)
\]

we obtain

\[
(\eta_1^{(1)}, \eta_2^{(1)}, \eta_3^{(1)}) = (0, -y', y - xy').
\]

Hence the determinant $D = \begin{vmatrix} \phi_i & 1 & 2x & x^2 \\ \eta_i^{(1)} & 0 & y & xy \\ \eta_i^{(2)} & 0 & -y' & y - xy' \end{vmatrix}$ equals $y^2$. So by Theorem 1 there does not exist any differential equation of order 1 which is invariant under the group of symmetries corresponding to the Lie algebra $g_2$ given by (32).

To determine all ordinary differential equations of order $\geq 2$ which are invariant under the 3-dimensional group of symmetries corresponding to the Lie algebra $g_2$ we need to compute the 3-th prolonged vector fields $X_i^{(3)}$, $i = 1, 2, 3$, of (32). Using (33) we obtain

\[
(\eta_1^{(1)}, \eta_2^{(1)}, \eta_3^{(1)}) = (0, -y', y - xy'),
\]

\[
(\eta_1^{(2)}, \eta_2^{(2)}, \eta_3^{(2)}) = (0, -3y^{(2)}, -3xy^{(2)}),
\]

\[
(\eta_1^{(3)}, \eta_2^{(3)}, \eta_3^{(3)}) = (0, -5y^{(3)}, -3y^{(2)} - 5xy^{(3)}).
\]
Therefore we have to solve the following system of partial differential equations:

\[
\begin{align*}
2x \frac{\partial f}{\partial x} + y \frac{\partial f}{\partial y} - y \frac{\partial f}{\partial y'} - 3y^{(2)} \frac{\partial f}{\partial y^{(2)}} - 5y^{(3)} \frac{\partial f}{\partial y^{(3)}} &= 0, \\
x^2 \frac{\partial f}{\partial x} + xy \frac{\partial f}{\partial y} + (y - xy') \frac{\partial f}{\partial y'} - 3xy^{(2)} \frac{\partial f}{\partial y^{(2)}} - (3y^{(2)} + 5xy^{(3)}) \frac{\partial f}{\partial y^{(3)}} &= 0.
\end{align*}
\] (33)

From the first equation of (33) it follows that \( f \) does not depend on the variable \( x \). Using this, system (33) reduces to the following:

\[
\begin{align*}
y \frac{\partial f}{\partial y} - y' \frac{\partial f}{\partial y'} - 3y^{(2)} \frac{\partial f}{\partial y^{(2)}} - 5y^{(3)} \frac{\partial f}{\partial y^{(3)}} &= 0, \\
y \frac{\partial f}{\partial y'} - 3y^{(2)} \frac{\partial f}{\partial y^{(3)}} &= 0.
\end{align*}
\] (34)

The solutions of the second equation are generated by

\[
y, \; y^{(2)}, \; \beta = \frac{3y^{(2)}y'}{y} + y^{(3)}.
\] (35)

Taking the functions in (35) as new variables the vector field \( Y = y \frac{\partial}{\partial y} - y' \frac{\partial}{\partial y'} - 3y^{(2)} \frac{\partial}{\partial y^{(2)}} - 5y^{(3)} \frac{\partial}{\partial y^{(3)}} \) can be written into the form:

\[
\begin{align*}
\bar{Y} &= Y(y) \frac{\partial}{\partial y} + Y(y^{(2)}) \frac{\partial}{\partial y^{(2)}} + Y(\beta) \frac{\partial}{\partial \beta} = \\
&= y \frac{\partial}{\partial y} - 3y^{(2)} \frac{\partial}{\partial y^{(2)}} - 5\beta \frac{\partial}{\partial \beta}.
\end{align*}
\]
Using integration, the solutions of the partial differential equation
\[ y \frac{\partial f}{\partial y} - 3y^{(2)} \frac{\partial f}{\partial y^{(2)}} - 5\beta \frac{\partial f}{\partial \beta} = 0 \]
are generated by
\[ \phi_1 = y^3 y^{(2)}, \quad \phi_2 = \beta y^5 = y^4 (3y^{(2)}y' + y^{(3)}y). \]

According to Theorem 2 any ordinary differential equation of order \( m \geq 2 \) which is invariant under the action of \( g_2 \) has the form:
\[ \Omega(\phi_1, \phi_2, \phi_3, \ldots) = 0, \]
where \( \phi_1, \phi_2 \) is given by \((36)\).

The Lie algebra \( g_3 = sl_2(\mathbb{R}) \) is generated by the vector fields:
\[ (37) \quad X_1 = \frac{\partial}{\partial y}, \quad X_2 = y \frac{\partial}{\partial y}, \quad X_3 = y^2 \frac{\partial}{\partial y}. \]

Because one has
\[ (38) \quad (\phi_1, \phi_2, \phi_3) = (0, 0, 0), \quad (\eta_1, \eta_2, \eta_3) = (1, y, y^2). \]

Using \((3)\) we obtain \((\eta_1^{(1)}, \eta_2^{(1)}, \eta_3^{(1)}) = (0, y', 2yy')\). In this case the determinant \( D = \begin{vmatrix} \phi_i \\ \eta_i \\ \eta_i^{(1)} \end{vmatrix} = \begin{vmatrix} 0 & 0 & 0 \\ 1 & y & y^2 \\ 0 & -y' & 2yy' \end{vmatrix} \) is identically 0.

As the coefficient of \( \frac{\partial}{\partial y} \) in all vector fields given by \((37)\) depends solely on \( y \) the differential equation \( y' = 0 \) is invariant under the group \( G \) corresponding to the Lie algebra \( g_3 \).
To determine all ordinary differential equations of order $\geq 2$ which are invariant under the 3-dimensional group of symmetries corresponding to the Lie algebra $g_3$ we need to compute the 3-th prolonged vector fields $X^{(3)}_i$ of (37). Using (38) we get 

\[
(\eta_1^{(1)}, \eta_2^{(1)}, \eta_3^{(1)}) = (0, y', 2yy'),
\]

(39) 

\[
(\eta_1^{(2)}, \eta_2^{(2)}, \eta_3^{(2)}) = (0, y'^2 + 2yy'^2),
\]

(40) 

\[
(\eta_1^{(3)}, \eta_2^{(3)}, \eta_3^{(3)}) = (0, y'^3 + 6y'y'^2 + 2yy'^3).
\]

Hence we obtain the following system of partial differential equations:

\[
y' \frac{\partial f}{\partial y} + y \frac{\partial f}{\partial y'} + y^{(2)} \frac{\partial f}{\partial y'^{(2)}} + y^{(3)} \frac{\partial f}{\partial y'^{(3)}} = 0,
\]

(41) 

\[
y^2 \frac{\partial f}{\partial y} + 2yy' \frac{\partial f}{\partial y'} + 2((y')^2 + yy'^2) \frac{\partial f}{\partial y'^{(3)}} + 2(3y'y'^2 + yy'^3) \frac{\partial f}{\partial y'^{(3)}} = 0.
\]

From the first equation of (41) it follows that $f$ does not depend on the variable $y$. Using this, system (41) reduces to the following:

\[
y' \frac{\partial f}{\partial y'} + y^{(2)} \frac{\partial f}{\partial y'^{(2)}} + y^{(3)} \frac{\partial f}{\partial y'^{(3)}} = 0.
\]

(42) 

\[
(y')^2 \frac{\partial f}{\partial y'^{(2)}} + 3y'y'^2 \frac{\partial f}{\partial y'^{(3)}} = 0.
\]
By integration of the second equation we obtain that its solutions are generated by:

$$x, \ y', \ \kappa = \frac{3}{2}(y^{(2)})^2 - y'y^{(3)}.$$  \hspace{1cm} (43)

Taking the functions in (43) as new variables the vector field

$$X_2^{(3)} = y' \frac{\partial}{\partial y'} + \cdots + y^{(3)} \frac{\partial}{\partial y^{(3)}}$$

can be written into the form:

$$X_2^{(3)}(y') \frac{\partial}{\partial y'} + X_2^{(3)}(\kappa) \frac{\partial}{\partial \kappa} = y' \frac{\partial}{\partial y'} + 2\kappa \frac{\partial}{\partial \kappa}. \hspace{1cm} (44)$$

Using integration, the solutions of the partial differential equation

$$y' \frac{\partial f}{\partial y'} + 2\kappa \frac{\partial f}{\partial \kappa} = 0$$

are generated by

$$\varphi_1 = x, \ \varphi_2 = \frac{\kappa}{(y')^2} = \frac{3(y^{(2)})^2 - 2y'y^{(3)}}{2(y')^2}.$$ 

Moreover, one has

$$\varphi_3 = \frac{d\varphi_2}{dx} = \frac{\beta}{(y')^2} = \frac{(y')^2 y^{(4)} + 3(y^{(2)})^2 - 4y'y^{(2)}y^{(3)}}{(y')^2}$$

and so further. Therefore, any ordinary differential equation of order $m \geq 2$ which is invariant under the action of $g_3$ has order at least 3 and takes the form:

$$\Omega(x, \varphi_2, \frac{d\varphi_2}{dx}, \frac{d^2\varphi_2}{dx^2}, \cdots) = 0,$$

where $\varphi_2 = \frac{3(y^{(2)})^2 - 2y'y^{(3)}}{2(y')^2}$ (cf. Theorem 2).
Lie algebra $\mathfrak{g}_{4} = \mathfrak{sl}_2(\mathbb{R})$ is generated by the vector fields:

\[
X_1 = \frac{\partial}{\partial x}, \quad X_2 = x \frac{\partial}{\partial x} + y \frac{\partial}{\partial y}, \quad X_3 = (x^2 - y^2) \frac{\partial}{\partial x} + 2xy \frac{\partial}{\partial y}.
\]

This is the only one representation of the Lie algebra $\mathfrak{sl}_2(\mathbb{R})$ such that the corresponding group action is primitive on the plane. Since one has

\[
(\phi_1, \phi_2, \phi_3) = (1, x, x^2 - y^2), (\eta_1, \eta_2, \eta_3) = (0, y, 2xy).
\]

Applying (3) we obtain

\[
(\eta_1^{(1)}, \eta_2^{(1)}, \eta_3^{(1)}) = (0, 0, 2y(1 + (y')^2)).
\]

Hence the determinant $D = \begin{vmatrix} \phi_i & 1 & x & x^2 - y^2 \\ \eta_i & 0 & y & 2xy \\ \eta_i^{(1)} & 0 & 0 & 2y(1 + (y')^2) \end{vmatrix}$ equals $2y^2(1 + (y')^2)$, according to Theorem 1 there does not exist any first order differential equation which admits a Lie group of symmetries having the Lie algebra $\mathfrak{g}_{4}$ as its Lie algebra.
To determine all ordinary differential equations of order $\geq 2$ which are invariant under the 3-dimensional group of symmetries corresponding to the Lie algebra $\mathfrak{g}$ we need to compute the 3-th prolonged vector fields $X^3_j$ of (45). Applying (46) we obtain

$$(\eta_1^{(1)}, \eta_2^{(1)}, \eta_3^{(1)}) = (0, 0, 2y(1 + (y')^2)),
$$

$$(\eta_1^{(2)}, \eta_2^{(2)}, \eta_3^{(2)}) = (0, -y^{(2)}, 2y' + 2(y')^3 + 6yy'y^{(2)} - 2xy^{(2)}),
$$

$$(\eta_1^{(3)}, \eta_2^{(3)}, \eta_3^{(3)}) = (0, -2y^{(3)}, 12(y')^2y^{(2)} + 6y(y^{(2)})^2 + 8yy'y^{(3)}).
$$

Hence we have the following system of partial differential equations:

$$
\begin{align*}
\frac{\partial f}{\partial x} + y \frac{\partial f}{\partial y} - y^{(2)} \frac{\partial f}{\partial y^{(2)}} - 2y^{(3)} \frac{\partial f}{\partial y^{(3)}} &= 0, \\
(x^2 - y^{(2)}) \frac{\partial f}{\partial x} + 2xy \frac{\partial f}{\partial y} + 2y(1 + (y')^2) \frac{\partial f}{\partial y'} + 2(y' + (y')^3 + 3yy'y^{(2)} - xy^{(2)}) \frac{\partial f}{\partial y^{(2)}} + 2(6(y')^2y^{(2)} + 3y(y^{(2)})^2 + (4yy' - 2x)y^{(3)}) \frac{\partial f}{\partial y^{(3)}} &= 0.
\end{align*}
$$

(48)

From the first equation of (48) it follows that $f$ does not depend on the variable $x$. Using this, system (48) reduces to the following:

$$
\begin{align*}
y \frac{\partial f}{\partial y} - y^{(2)} \frac{\partial f}{\partial y^{(2)}} - 2y^{(3)} \frac{\partial f}{\partial y^{(3)}} &= 0, \\
2y(1 + (y')^2) \frac{\partial f}{\partial y'} + 2(y' + (y')^3 + 3yy'y^{(2)}) \frac{\partial f}{\partial y^{(2)}} + (12(y')^2y^{(2)} + 6y(y^{(2)})^2 + 8yy'y^{(3)}) \frac{\partial f}{\partial y^{(3)}} &= 0.
\end{align*}
$$

(49)
By integration of the first equation we obtain that its solutions are generated by:

\[(50)\quad y', \quad \alpha = y^{(2)}y, \quad \beta = y^{(3)}y^2.\]

Taking the functions in (50) as new variables the vector field \(Y = 2y(1 + (y')^2)\frac{\partial}{\partial y'} + 2(y' + (y')^3 + 3yy'y^{(2)})\frac{\partial}{\partial y} + \frac{(12(y')^2y^{(2)} + 6y(y^{(2)})^2 + 8yy'y^{(3)})\partial}{\partial y^{(3)}}\) can be written into the form:

\[(51)\quad \tilde{Y} = (1 + (y')^2)\frac{\partial}{\partial y'} + y'(1 + (y')^2 + 3\alpha)\frac{\partial}{\partial \alpha} + (6(y')^2\alpha + 3\alpha^2 + 4y'\beta)\frac{\partial}{\partial \beta}.

Using integration, the solutions of the partial differential equation

\[(1 + (y')^2)\frac{\partial f}{\partial y'} + y'(1 + (y')^2 + 3\alpha)\frac{\partial f}{\partial \alpha} + (6(y')^2\alpha + 3\alpha^2 + 4y'\beta)\frac{\partial f}{\partial \beta} = 0\]

are generated by

\[(52)\quad \varphi_1 = \frac{\alpha + 1 + (y')^2}{(1 + (y')^2)^{\frac{3}{2}}} = \frac{y^{(2)}y + 1 + (y')^2}{(1 + (y')^2)^{\frac{3}{2}}},\]

\[\varphi_2 = \frac{\beta(y')^2 + \beta - 3y'\alpha^2}{(1 + (y')^2)^3} = \frac{y^{(3)}y^2(1 + (y')^2) - 3y'(y^{(2)})^2y^2}{(1 + (y')^2)^3}.\]

Hence any ordinary differential equation of order \(m \geq 2\) which is invariant under the action of \(g_4\) takes the form:

\[\Omega(\varphi_1, \varphi_2, \cdots) = 0,\]

where \(\varphi_i, i = 1, 2,\) is given by (52) (cf. Theorem 2).
Acknowledgement

The research leading to these results has received funding from the European Union’s Seventh Framework Programme (FP7/2007-2013) under grant agreements no. 318202 and no. 317721.
Coupling as Colimit

Fariba Karimi and Chrystopher L. Nehaniv

Wolfson Royal Society Biocomputation Laboratory
Science and Technology Research Institute
University of Hertfordshire, U.K.
EU FP7 Project BIOMICS

3rd International BIOMICS Workshop,
Passau, Germany – February 8th-10th, 2016

NB: The description of coproduct of automata has been corrected (March 2016).

Outline

Colimits (Coproduct and Pushout) in the categories of

- Digraphs
- Permutation Groups
- ODE Systems
- Interaction Machines
Pushout of $\alpha$ and $\beta$

Pushout in Labelled Digraph Category

$\{v_2, w_2\}$

$\{v_1, w_1\}$

Pushout in Labelled Multi-Digraph Category

$\{v_2, w_2\}$

$\gamma$

$\delta$

$\{v_1, w_1\}$
Motivation: Gluing Faithful Permutation Groups and Transformation Semigroups

We show the existence and describe the structure of coproducts in the following categories with objects \((X, S)\), with \(X\) a set and \(S \subseteq X^X\) a set of functions on \(X\) closed under composition, writing \(x \cdot s\) for \(s \in S\) applied to \(x \in X\):

- permutation groups \(\text{PermGrp}\) (each \(s \in S\) is a permutation of \(X\) and \(S\) is group)
- transformation monoids \(\text{TM}\) (identity \(id_X \in S\))
- transformation semigroups \(\text{TS}\)
- partial transformation semigroups \(\text{PTS}\). (Each \(s\) a partial function from \(X\) to \(X\))

Also for the variants \(\text{PermGrp}_*, \text{TM}_*, \text{TS}_*, \text{PTS}_*\) of these categories with base-points \(* \in X\) and base-point preserving maps.

All in all these categories actions are faithful: if elements \(s_1, s_2\) of the group (resp., monoid, semigroup) act the same on all states, then they are equal.

Morphisms

- A morphism \(\psi\) of permutation groups \((X, S)\) to \((X', S')\) is a set map \(\psi^{\text{state}} : X \to X'\) and homomorphism \(\psi^{\text{operators}} : S \to S'\), with

\[
\psi^{\text{state}}(x \cdot s) = \psi^{\text{state}}(x) \cdot \psi^{\text{operators}}(s) \quad \forall x \in X, s \in S
\]

\[
\psi^{\text{operators}}(s_1 s_2) = \psi^{\text{operators}}(s_1) \psi^{\text{operators}}(s_2) \quad \forall s_1, s_2 \in S
\]

It follows that inverses map to inverses, and identity of \(S\) maps to identity element of \(S'\) (since idempotents map to idempotents).

- A transformation semigroup morphism is defined the same way. For the transformation monoid category, one must require of morphisms, that the identity of \(S\) map to that of \(S'\).
Coproduct of Groups, Monoids, or Semigroups

In groups or monoids, the coproduct is the “free product”.

\[ S \ast T = \{ (a_1, \ldots, a_k) : k \geq 0, \text{ with the } a_i \neq 1 \text{ alternating membership in } S \text{ and } T \}. \]

If \( k = 0 \) this is the identity element of \( S \ast T \).

Multiply: \((a_1, \ldots, a_k)(b_1, \ldots, b_n) =
\begin{cases} 
(a_1, \ldots, a_k, b_1, \ldots, b_n) & \text{if } a_k \in S, b_1 \in T, \text{ or } a_k \in T, b_1 \in S \\
\text{reduce}(a_1, \ldots, a_k b_1, \ldots, b_n) & \text{if } a_k, b_1 \in S \text{ or } a_k, b_1 \in T
\end{cases}
\]

where \text{reduce} means removing any 1’s that appear, and combine any new neighbors by multiplication if both are from same \( S \) or \( T \), and then iterating reduction to get a canonical form.

Coproduct of Semigroups

\[ S \ast T = \{ (a_1, \ldots, a_k) : k \geq 1, \text{ with the } a_i \text{ alternating membership in } S \text{ and } T \}. \]

Multiply: \((a_1, \ldots, a_k)(b_1, \ldots, b_n) =
\begin{cases} 
(a_1, \ldots, a_k, b_1, \ldots, b_n) & \text{if } a_k \in S, b_1 \in T, \text{ or } a_k \in T, b_1 \in S \\
(a_1, \ldots, a_k b_1, \ldots, b_n) & \text{if } a_k, b_1 \in S \text{ or } a_k, b_1 \in T
\end{cases}
\]

- For semigroups, coproduct of two nonempty semigroups is always infinite, e.g., \( 1 \ast 1 \) is infinite.

\[ S \ast T = T \ast S, \quad \emptyset \ast S = S, \]
A coproduct \((X, S) \coprod (Y, T)\) of permutation groups \((X, S)\) and \((Y, T)\), if it exists is some \((Q, C)\) with two maps \(i_{(X,S)}\) and \(i_{(Y,T)}\) to \((Q, C)\) such that when \(j\)'s are given to some permutation group \((Z, U)\) then these factor uniquely through \((Q, C)\):

\[
(X, S) \quad \xrightarrow{i_{(X,S)}} \quad (Q, C) \quad \xrightarrow{j_{(X,S)}} \quad (Z, U) \\
\downarrow j_{(X,S)} \hspace{2cm} \exists! \varphi \hspace{2cm} \downarrow \exists! j_{(Y,T)} \\
(Y, T) \quad \xrightarrow{i_{(Y,T)}} \quad (Q, C) \quad \xrightarrow{j_{(Y,T)}} \quad (Z, U)
\]

Observe: A coproduct is unique up to isomorphism (if it exists).

**Coproduct of Permutation Groups**

**Theorem**

*In the category of permutation groups \(\text{PermGrp}\), given permutation groups \((X, S)\) and \((Y, T)\), their coproduct is*

\[
((X \sqcup Y) \otimes (S * T), S * T),
\]

where \(S * T\) is the free product of groups and \((X \sqcup Y) \otimes (S * T)\) denotes \(((X \sqcup Y) \times (S * T))/\equiv\) under the equivalence relation \(\equiv\) generated by

\[
(a, sw) \sim (a \cdot s, w), \quad \text{if} \ a \in X, s \in S, \\
(a, tw) \sim (a \cdot t, w), \quad \text{if} \ a \in Y, t \in T,
\]

where \(i_X : (X, S) \to ((X \sqcup Y) \otimes (S * T), S * T)\) maps \(x \mapsto (x, 1), s \mapsto s \in S * T\), and \(i_Y : (Y, T) \to ((X \sqcup Y) \otimes (S * T), S * T)\) maps \(y \mapsto (y, 1), t \mapsto t \in S * T\).
Outline of Proof for Coproduct of Permutation Groups

Rewrite elements of \((X \sqcup Y) \times (S \ast T)\) to equivalent elements in canonical form, \(a \in X \sqcup Y, w \in S \ast T\): Move letters \(s\) and \(t\) from \(w\) to the left when action of the \(s\) or \(t\) is defined on \(a\) in the factor until impossible. Action \((a, w) \cdot w' = (a, ww')\), \(a \in X \sqcup Y, w, w' \in S \ast T\) is well-defined on equivalence classes.

Action is faithful, so we have a (faithful) permutation group \(((X \sqcup Y) \otimes (S \ast T), S \ast T)\).

Existence of unique morphism to any \((Z, U)\) making diagram commute: let \(\varphi : S \ast T \to U\) be the unique homomorphism (for the coproduct \(S \ast T\)).

For states: Map \((a, w)\) to \(j_x(a) \cdot \varphi(w)\) if \(a \in X\) or to \(j_y(a) \cdot \varphi(w)\) for \(a \in Y\).

This is well-defined on equivalence classes: if we apply an equivalence rule this gives same member of \(Z\). E.g.,

\[(x, sw) \mapsto j_x(x) \cdot \varphi(sw) = j_x(x) \cdot \varphi(s) \varphi(w) = (j_x(x) \cdot \varphi(s)) \cdot \varphi(w) = (j_x(x) \cdot j_s(s)) \cdot \varphi(w) = j_x(x \ast s) \cdot \varphi(w),\]

which is where \((x, sw)\) maps.

The diagram commutes as required. Uniqueness of state-map follows easily since it is determined where \((x, 1)\) and \((y, 1)\) must go, and hence where \((x, w) = (x, 1) \cdot w\) and \((y, w) = (y, 1) \cdot w\) go. \(\square\)

Theorems for Coproducts of Transformation Monoids & Semigroups, and with basepoints and/or parital

- Coproducts for transformation monoids are constructed in exactly the same way.
- Coproducts of transformation semigroups are constructed the same way, semigroup acting is \(S \ast T\), but for states \(S \ast T\) in \((X \sqcup Y) \times (S \ast T)\) is augmented to \((S \ast T) \cup \{\lambda\}\), where \(\lambda\) in 2nd coordinate serves the same role 1 did in the permutation group case. (Works for \(|X|, |Y| \geq 1\).)
Theorems for Coproducts of Transformation Monoids & Semigroups, and with basepoints and/or parital

- Coproducts for transformation monoids are constructed in exactly the same way.
- Coproducts of transformation semigroups are constructed the same way, semigroup acting is $S \star T$, but for states $S \star T$ in $(X \sqcup Y) \times (S \star T)$ is augmented to $(S \star T) \cup \{\lambda\}$, where $\lambda$ in 2nd coordinate serves the same role 1 did in the permutation group case. (Works for $|X|, |Y| \geq 1$.)
- With basepoints, one also obtains a canonical form for states, adding one more rule $(x_0, w) \sim (y_0, w)$, where $x_0$ is the basepoint of $X$, $y_0$ is the basepoint of $Y$, and $w \in S \star T$. In the semigroup case, we allow $w = \lambda$, the empty word. Coproduct exists for $|X|, |Y| > 1$.

Theorems for Coproducts of Transformation Monoids & Semigroups, and with basepoints and/or parital

- Coproducts for transformation monoids are constructed in exactly the same way.
- Coproducts of transformation semigroups are constructed the same way, semigroup acting is $S \star T$, but for states $S \star T$ in $(X \sqcup Y) \times (S \star T)$ is augmented to $(S \star T) \cup \{\lambda\}$, where $\lambda$ in 2nd coordinate serves the same role 1 did in the permutation group case. (Works for $|X|, |Y| \geq 1$.)
- With basepoints, one also obtains a canonical form for states, adding one more rule $(x_0, w) \sim (y_0, w)$, where $x_0$ is the basepoint of $X$, $y_0$ is the basepoint of $Y$, and $w \in S \star T$. In the semigroup case, we allow $w = \lambda$, the empty word. Coproduct exists for $|X|, |Y| > 1$.
- For partial transformation semigroups, coproduct is very different. States are just the disjoint union. Semigroup acting is just union of $S$ and $T$ which are undefined if they act on the state of the other component. Similarly for partial transformation semigroups with basepoint.
Theorems for Coproducts of Automata [Corrected Slide]

- **Theorem (Coproduct for Automata).** For complete deterministic reachable automata with initial state, the states of the coproduct of automata \( A = (Q_A, X, i_A, \delta_X : Q_A \times X \rightarrow Q_A) \) and \( B = (Q_B, Y, i_B, \delta_Y : Q_B \times Y \rightarrow Q_B) \) are: \( (Q_A \sqcup Q_B) \otimes (X^* \times Y^*)^\lambda = (Q_A \sqcup Q_B) \otimes (X \sqcup Y)^* \), with action as in the coproduct of pointed transformation semigroups, taking the initial states as basepoints. That is, the coproduct is the complete deterministic reachable automaton, \( A \sqcup B = ((Q_A \sqcup Q_B) \otimes (X \sqcup Y)^*, X \sqcup Y, i, \delta) \), with initial state \( i = i_X \otimes \lambda = i_Y \otimes \lambda \), \( \delta(q \otimes w, z) = q \otimes wz \) for all \( q \in Q_A \sqcup Q_B, w \in (X \sqcup Y)^*, z \in X \sqcup Y \). Here \( a \otimes x = \delta_A(a, x) \otimes \lambda \) and \( b \otimes y = \delta_B(b, y) \otimes \lambda \) for all \( a \in Q_A, x \in X, b \in Q_B, y \in Y \); but \( a \otimes yw \) and \( b \otimes xw \) can never be ‘reduced’ unless \( a = i_A \) or \( b = i_B \).

- For partial automata, just put the automata next to each other, identifying their initial states, use disjoint union of input alphabets.

- For nondeterministic partial automata, the states are as for partial transformation semigroups, initial states identified, and input alphabet the disjoint union of the input alphabets.

Category of ODE Systems

- **An object in the category of time-dependent first-order ODEs** is defined as a system of equations:

\[
\frac{dx_1}{dt} = g(x_1, \ldots, x_n) \\
\vdots \\
\frac{dx_n}{dt} = g_n(x_1, \ldots, x_n),
\]

which can be summarized as

\[
\frac{d\vec{x}}{dt} = \vec{g}(\vec{x}, t).
\]
Morphism in the Category of ODE Systems

- An $\alpha$-morphism of systems of ODEs $\sigma: \Sigma \rightarrow \Sigma'$, where $\Sigma$ has $n$ variables and $\Sigma'$ has $m$ variables, is a substitution of variables, e.g., an algebraic expression in the variables of $\Sigma'$, or more generally a continuously differentiable function of the variables $(y_1, \ldots, y_m)$ from $\Sigma'$, for each of the variables $x_i$ of $\Sigma$, i.e.,

$$x_i = \sigma_i(y_1, \ldots, y_m)$$

such that for every equation $e$ in the variables and their derivatives of $\Sigma$, with these substitutions $\sigma(e)$ holds in $\Sigma'$.

- Here, $\sigma(e)$ denotes the equation over the variables of $\Sigma'$ resulting from replacing the variables from $\Sigma$ in equation $e$ according to the substitution $\sigma$.

- Two $\alpha$-morphisms $\sigma, \sigma': \Sigma \rightarrow \Xi$ are equal if and only if for all variables $x$ of $\Sigma$, $\sigma(x) = \sigma'(x)$ holds in $\Xi$ (i.e., as a consequence of the equations of $\Xi$ (without using any equations of $\Sigma$)).

Examples of $\alpha$-Morphism

- $\dot{w}_1 = 2w_1 + 2w_2$\\
  $f \begin{cases} w_1 = w_1 \\ w_2 = w_2 \end{cases}$\quad $\frac{\dot{w}_1 + \dot{w}_2}{\dot{w}_1 - \dot{w}_2} = \frac{4w_1 + 2w_2}{2w_2}$
Examples of $\alpha$-Morphism

\[
\begin{align*}
\dot{w}_1 &= 2w_1 + 2w_2 \\
\dot{w}_2 &= w_1 \\
\dot{w}_1 - \dot{w}_2 &= 2w_2
\end{align*}
\]

\[
\begin{align*}
\dot{w}_1 + \dot{w}_2 &= 4w_1 + 2w_2 \\
\dot{w}_1 - \dot{w}_2 &= 2w_2 \\
\Rightarrow 2\dot{w}_1 &= 4w_1 + 4w_2 \\
\Rightarrow \dot{w}_1 &= 2w_1 + 2w_2
\end{align*}
\]
Examples of $\alpha$-Morphism

Given:

\[
\begin{align*}
\dot{w}_1 &= 2w_1 + 2w_2 \\
\dot{w}_2 &= w_1 = w_2
\end{align*}
\]

Then:

\[
\begin{align*}
\dot{w}_1 + \dot{w}_2 &= 4w_1 + 2w_2 \\
\dot{w}_1 - \dot{w}_2 &= 2w_2
\end{align*}
\]

\[\Rightarrow 2\dot{w}_1 = 4w_1 + 4w_2 \Rightarrow \dot{w}_1 = 2w_1 + 2w_2\]

Additional Example:

\[
\begin{align*}
\dot{w}_1 + \dot{w}_2 &= 4w_1 + 2w_2 \\
\dot{w}_1 - \dot{w}_2 &= 2w_2
\end{align*}
\]

\[\Rightarrow \dot{w}_1 = 2w_1 + 2w_2\]
Examples of $\alpha$-Morphism

- $\dot{w}_1 = 2w_1 + 2w_2 \quad \begin{cases} w_1 = w_1 \\ w_2 = w_2 \end{cases} \quad \Rightarrow \dot{w}_1 + \dot{w}_2 = 4w_1 + 2w_2 \quad \checkmark$
  \[ \dot{w}_1 - \dot{w}_2 = 2w_2 \]

  \[ \dot{w}_1 = 2w_1 + 2w_2 \Rightarrow 2\dot{w}_1 = 4w_1 + 4w_2 \Rightarrow \dot{w}_1 = 2w_1 + 2w_2 \]

- $\dot{w}_1 + \dot{w}_2 = 4w_1 + 2w_2 \quad \begin{cases} w_1 = w_1 \\ w_2 = w_2 \end{cases} \quad \Rightarrow \dot{w}_1 = 2w_1 + 2w_2 \quad \times$

  \[ \dot{w}_1 = 2w_1 + 2w_2 \neq \dot{w}_1 + \dot{w}_2 = 4w_1 + 2w_2 \]

Example of Equality of Two $\alpha$-Morphisms

- Is $f = g$?

  \[ \begin{cases} \dot{w}_1 = 2w_1 + 2w_2 \\ \dot{w}_1 = \dot{x}_1 + \dot{x}_2 \\ \dot{w}_2 = \dot{x}_1 - \dot{x}_2 \end{cases} \Rightarrow \begin{cases} x_1 = 3x_1 + x_2 \\ x_2 = x_1 - x_2 \end{cases} \]

(5)
Example of Equality of Two $\alpha$-Morphisms

- Is $f = g$? No

\[
\dot{w}_1 = 2w_1 + 2w_2 \quad \Rightarrow \quad \begin{cases} 
  w_1 &= x_1 + x_2 \\
  w_2 &= x_1 - x_2 
\end{cases} \quad \begin{cases} 
  \dot{x}_1 &= 3x_1 + x_2 \\
  \dot{x}_2 &= x_1 - x_2 
\end{cases}
\]

\[ (5) \]

- Is $f_1 = g_1$?

\[
\dot{w}_1 = 2w_1 + 2w_2 \quad \Rightarrow \quad \begin{cases} 
  w_1 &= x_1 + x_2 \\
  w_2 &= x_1 - x_2 
\end{cases} \quad \begin{cases} 
  \dot{x}_1 &= 3x_1 + x_2 \\
  \dot{x}_2 &= x_1 - x_2 \\
  x_1 &= x_2 = 0 
\end{cases}
\]

\[ (6) \]
Example of Equality of Two $\alpha$-Morphisms

- Is $f = g$? No

\[
\begin{align*}
\dot{w}_1 &= 2w_1 + 2w_2 \\
\dot{w}_2 &= \begin{cases}
w_1 = x_1 + x_2 \\
w_2 = x_1 - x_2 \end{cases} \\
\dot{x}_1 &= 3x_1 + x_2 \\
\dot{x}_2 &= x_1 - x_2 \\
f \begin{cases} w_1 = \frac{x_1 + x_2}{2} \\
w_2 = \frac{x_1 - x_2}{2} \end{cases} \\
g \begin{cases} w_1 = \frac{x_1 + x_2}{2} \\
w_2 = \frac{x_1 - x_2}{2} \end{cases}
\end{align*}
\]  

- Is $f_1 = g_1$? Yes

\[
\begin{align*}
\dot{w}_1 &= 2w_1 + 2w_2 \\
\dot{w}_2 &= \begin{cases}
w_1 = x_1 + x_2 \\
w_2 = x_1 - x_2 \end{cases} \\
\dot{x}_1 &= 3x_1 + x_2 \\
\dot{x}_2 &= x_1 - x_2 \\
f \begin{cases} w_1 = \frac{x_1 + x_2}{2} \\
w_2 = \frac{x_1 - x_2}{2} \end{cases} \\
g_1 \begin{cases} w_1 = \frac{x_1 + x_2}{2} \\
w_2 = \frac{x_1 - x_2}{2} \end{cases}
\end{align*}
\]

Pushout of ODE Systems

\[
\begin{array}{c}
\Sigma_X \\
\uparrow \sigma_X \\
\Sigma_Y \\
\uparrow \sigma_Y \\
\Sigma_Z \\
\downarrow \sigma_Z \\
\Sigma_Z \\
\end{array}
\]
Pushout of ODE Systems
Theorem (Pushout of ODE Systems)

Let $\Sigma_X: \frac{dx}{dt} = \vec{f}(\vec{x})$, $\Sigma_Y: \frac{dy}{dt} = \vec{g}(\vec{y})$, $\Sigma_Z: \frac{dz}{dt} = \vec{h}(\vec{z})$ be three systems of differential equations, $\sigma_X: \Sigma_Y \to \Sigma_X$ and $\sigma_Z: \Sigma_Y \to \Sigma_Z$ be two $\alpha$-morphisms. Define the ODE system $\Sigma_P$ by the disjoint union of the following systems,

$$
\begin{align*}
\frac{d\vec{x}}{dt} &= \vec{f}(\vec{x}) \\
\frac{d\vec{z}}{dt} &= \vec{h}(\vec{z}) \\
\sigma_X(\vec{x}) &= \sigma_z(\vec{z})
\end{align*}
$$

(7)

and two $\alpha$-morphisms $i_X: \Sigma_X \to \Sigma_P$ and $i_Z: \Sigma_Z \to \Sigma_P$ by

$$
\begin{align*}
i_X(\vec{x}) &= \vec{x}, \\
i_Y(\vec{y}) &= \vec{y}.
\end{align*}
$$

(8) (9)

Then, the ODE system $\Sigma_P$ with the morphisms $i_X$ and $i_Z$ is the pushout of $\sigma_X$ and $\sigma_Z$ in the category of $\mathcal{ODE}$. 
Example

\[
  f: \left\{ \begin{array}{l}
  \psi_a = \psi_a \\
  \psi_b = \psi_b \\
  z = \frac{a}{m}(\psi_b - \psi_a)
  \end{array} \right. \\

  (z, \psi_a, \psi_b)
\]

\[
  \psi_a = -\frac{b}{m} \psi_a + z \\
  \psi_b = -\frac{b}{m} \psi_b - z \\
  z = \frac{m}{a}(\psi_b - \psi_a)
\]

\[
  \psi_1 = -\frac{1}{m} \psi_1 \\
  \psi_2 = -3 \frac{c}{m} \psi_2
\]

Coequalizer of ODE Systems

\[
\Sigma_1 \xrightarrow{f} \xrightarrow{g} \Sigma_2
\]
Coequalizer of ODE Systems

\[ \Sigma_1 \xrightarrow{f} \Sigma_2 \xrightarrow{g} \Sigma \]

\[ \downarrow \quad \downarrow \]

\[ u \quad ? \]
Theorem (Coequalizer of ODE systems)

Let \( f, g : \Sigma_1 \rightrightarrows \Sigma_2 \) be two \( \alpha \)-morphisms where the ODE systems \( \Sigma_1 \) and \( \Sigma_2 \) consist of variables \( x_1, \ldots, x_n \) and \( y_1, \ldots, y_m \), respectively. Also, \( f \) and \( g \) are defined by the substitutions \( x_i = \sigma_i^f(y_1, \ldots, y_m) \) and \( x_i = \sigma_i^g(y_1, \ldots, y_m) \), respectively. Then, the coequalizer of \( f \) and \( g \) is the inclusion \( \alpha \)-morphism \( u : \Sigma_2 \rightarrow \Sigma_2 \cup \Sigma_3 \) where \( \Sigma_3 \) is defined by the equations

\[
\sigma_i^f(y_1, \ldots, y_m) = \sigma_i^g(y_1, \ldots, y_m),
\]

for \( i = 1, \ldots, n \).

\[
\Sigma_1 \xrightarrow{f} \Sigma_2 \xrightarrow{g} \Sigma_2 \cup \Sigma_3 \xrightarrow{h} \Sigma
\]
Example of Coequalizer of ODE Systems

\[
\begin{align*}
\dot{w}_1 &= 2w_1 + 2w_2 \\
\begin{cases}
w_1 &= x_1 + x_2 \\
w_2 &= x_1 - x_2
\end{cases} \\
\begin{cases}
x_1 &= 3x_1 + x_2 \\
x_2 &= x_1 - x_2 \\
x_1 &= x_2 = 0
\end{cases}
\end{align*}
\]

(12)

Colimits of ODE Systems

Corollary

*The colimit of every finite diagram of ODE Systems exists.*
Coproduct of $\text{Im}.$-Systems

**Definition**

Let $A$ be the set of inputs, $B$ be the set of outputs, $V$ be a set of indexes, and $\mathcal{P}_{\text{fin}}(V)$ be the set of finite subsets of $V$. The category of Level 1 Interaction Machines (with local $D^*_\text{fin}$-systems) is defined by the functor $\text{Im}.$, where for set $X$,

$$\text{Im}.(X) = \left( \bigcup \sqcup D^*_\text{fin}\text{-systems} \right)^V \times \left( \left( \bigcup \sqcup B \right)^V \right)^A \times X^A. \quad (13)$$

An $\text{Im}.$-coalgebra $\alpha$ is a function $X \xrightarrow{\alpha} \text{Im}.(X)$ that maps an element $x$ in $X$ to the triple $(\Omega_x, E_x, \Delta_x)$ where $\Omega_x : V \to \left( \bigcup \sqcup D^*_\text{fin}\text{-systems} \right)$ is the topology at state $x$, $E : A \to \left( \bigcup \sqcup B \right)^V$ and $\Delta_x : A \to X$ are the output function and global transition functions at state $x \in X$, respectively.

**Theorem**

Let $(X, \alpha)$ and $(Y, \beta)$ be two $\text{Im}.$-systems. Then, the coproduct of $\alpha$ and $\beta$ is the $\text{Im}.$-system $\gamma : X \sqcup Y \to \text{Im}.(X \sqcup Y)$ defined by

$$\gamma(z) = \begin{cases} 
\left(\text{Im}.(i_X) \circ \alpha\right)(z), & \text{if } z \in X, \\
\left(\text{Im}.(i_Y) \circ \beta\right)(z), & \text{if } z \in Y,
\end{cases} \quad (14)$$

where $i_X : X \to X \sqcup Y$ and $i_Y : Y \to X \sqcup Y$ are inclusion functions.
Colimit of $\text{Im}$.-Systems

**Theorem (Coequalizer of $\text{Im}$.-Systems)**

Let $f, g : \langle X, \alpha \rangle \to \langle Y, \beta \rangle$ be two $\text{Im}$.-homomorphisms from $\text{Im}$.-system $\alpha$ to the $\text{Im}$.-system $\beta$ where for each $y \in Y$, $\beta(y) = (\Omega_y, E_y, \Delta_y)$. Then, the coequalizer of $f$ and $g$ is the $\text{Im}$.-homomorphism $u : \langle Y, \beta \rangle \to \langle Y/\sim_{f,g}, \kappa \rangle$ with the following construction: For every $y \in Y$, $u(y) = [y]$ and $\kappa : Y/\sim_{f,g} \to \text{Im}.(Y/\sim_{f,g})$ is the $\text{Im}$.-system defined by $\kappa([y]) = (\Omega_y, E_y, \widetilde{\Delta_y})$ where $[y] \in Y/\sim_{f,g}$ and $\Delta_y : A \to Y$ is defined by $\Delta_y(a) = [\Delta_y(a)]$ for every $a \in A$.

**Corollary**

The colimit of every finite $\text{Im}$.-diagram exists.

**Limit of $\text{Im}$.-diagrams**

**Definition**

A functor $F : \text{Set} \to \text{Set}$ is **bounded**, if there is a cardinality $\kappa$, so that for every $F$-coalgebra $\langle X, \alpha \rangle$ and any $x \in X$, there exists a subcoalgebra $\langle Y_x, \beta_x \rangle$ of $\alpha$ of cardinality at most $\kappa$ with $x \in Y_x$.

**Lemma**

$\text{Im}$. is a bounded functor.

**Theorem**

The limit of every finite $\text{Im}$.-diagram exists.

**Theorem**

Let $f, g : \langle X, \alpha \rangle \Rightarrow \langle Y, \beta \rangle$ be two $\text{Im}$.-homomorphisms from $\text{Im}$.-system $\alpha$ to the $\text{Im}$.-system $\beta$. Then, the equalizer of $f$ and $g$ is the $\text{Im}$.-homomorphism $e : \langle E, \gamma \rangle \to \langle X, \alpha \rangle$ where $E = \{x \in X : \alpha(x) = \beta(x)\}$, $\gamma : E \to \text{Im}.(E)$ is the restriction of $\alpha$ to $E$ and $e(x) = x$ for every $x \in X$. 
Appendix

14 How to avoid Petri net idiosyncrasies when modeling computational systems, by Egon Berger (Abstract)

We explain by characteristic examples taken from N. Lynch's book on "Distributed Algorithms" how to model distributed algorithms by Abstract State Machines (ASMs). It turns out that the used ASMs are networks of communicating ASMs, a class of sequential ASMs each of which works independently of the others (in particular with its own clock) and shares information with them only via communication along the topological network structure; to achieve generality the adopted communication mechanism and the topology are kept abstract so that they can be implemented by current communication systems. Comparing these communicating ASMs with Petri nets (PNs) reveals a certain number of idiosyncrasies of PNs which complicate both model design and analysis. The ASMs we define illustrate how one can avoid such framework related technicalities.

The details can be found in the paper “Modeling Distributed Algorithms by Abstract State Machines Compared to Petri Nets” In: M. Butler et al (Eds.): ABZ 2016, LNCS 9675, pp.1-32, 2016 DOI: 10:1007/978-3-319-33600-8.1
15 BSL: A CoreASM Modification for the BIOMICS Project
by Eric Rothstein Morris and Daniel Schreckling

BSL: a CoreASM Modification for the
BIOMICS Project

Eric Rothstein Morris, Daniel Schreckling
Institute of IT-Security and Security Law,
University of Passau, Passau, Germany
er@sec.uni-passau.de and ds@sec.uni-passau.de

Abstract. We introduce the BIOMICS Specification Language (BSL), a modification of CoreASM for the BIOMICS project. BSL provides new language primitives for the specification of behaviours observed in biological systems using Abstract State Machines (ASMs). These new BSL primitives allows users to precisely control the execution of multi-agent systems by defining scheduling policies that adapt as the program executes, and they also enhance CoreASM’s element selection mechanisms by allowing users to define under which probability distribution the choice action should be performed.

1 Introduction

Several biological systems exhibit behaviours that are interesting for software developers. In particular, cells show the ability to remain in dynamical stability; i.e., they can progressively recover their normality in spite of a wide variety of perturbations coming from their environment. Since the nature of biological systems is very different from the nature of software systems, we need mappings from biology to software engineering that translate the necessary conditions and components which promote the interesting behaviours that we observe. Creating correct models of these biological systems and their environment is one way to provide these mappings. A correct model does not necessarily capture all the biological aspects of a system; instead, it places emphasis on the important parts so that we can reproduce the behaviour we are interested in, and it abstracts the unnecessary details so that we can study the observed behaviour without unnecessary clutter.

Obtaining correct models of biological systems is not an easy task. The biologists that can explain the causes and conditions that promote certain behaviours usually do not define models with specification languages used in software engineering. Understandably, they provide many explanations for the causes and conditions in natural language, sometimes along with some sort of mathematical modelling for validation; e.g., a system of (approximated) ordinary differential equations (ODE). Moreover, the process of discovering the causes behind a particular behaviour observed in biology can be long and complicated; e.g., the calcium cycle of the heart consists of over 20 intertwined pathways, with
each pathway taking over 4 years to study independently. Consequently, successfully porting behaviours from biology to software engineering is unfortunately a lengthy, difficult process that is always prone to human error.

To close the gap between biology and software engineering so as to eventually have software that exhibits behaviours observed in biological systems, we propose using Abstract State Machines (ASMs) to correctly specify models of systems. The main advantage of ASMs over other specification languages is that they easily translate from natural language thanks to their rich syntax, which means that biologists do not have to think in terms of “tokens and locations” as if they were modelling systems using Petri Nets, or in terms of “finite states and transition inputs” as if they were using finite automata. For instance, we model the expression “whenever the concentration of calcium is higher than a given quantity \( c \), the muscle contracts” using the conditional rule

\[
\text{if } \text{calciumConcentration} > c \text{ then MuscleContracts}
\]

where the value \( \text{calciumConcentration} \), the value \( c \), and the rule \( \text{MuscleContracts} \) can be further refined.

In this work, we introduce the BIOMICS Specification Language (BSL) in an effort towards having a modelling language that lets biologists specify their models naturally using ASMs. Having ASM models of biological systems allows computer scientist to observe and experiment hands-on with the modelled systems, finding patterns that may be applicable to software engineering.

BSL is a modification of CoreASM, an open source project (licensed under Academic Free License version 3.0) that focuses on the design of a lean executable ASM language, in combination with a supporting tool environment for high-level design, experimental validation, and formal verification (where appropriate) of abstract system models [3]. We extended CoreASM because, during the BIOMICS project, we identified a couple of language primitives that were missing from CoreASM when trying to model cell metabolic pathways; thus, a subset of the CoreASM language plus the identified missing primitives is what we call BSL.

We assume the reader to be familiar with the ASM language described in the ASM Book [1]. We invite the reader unfamiliar with CoreASM to check the CoreASM user manual [2] as well as Deliverable 4.2 of BIOMICS [4].

### 2 Choosing using expected probabilities

The probability distribution background of the BSL is a new extension to CoreASM. Many phenomena in biology do not follow a uniform distribution (e.g., phenotypic distributions of phenotypes), and it would be very useful to have selection functions that follow particular probability distributions. However, CoreASM’s selection functions (both the choose rule and the pick instruction) are implemented using fair selection functions that follow uniform distributions. Therefore, we extend CoreASM by introducing the notion of probability distributions, and the ability to choose using a probability distribution as a reference.
In general terms, a user-defined probability distribution is a map from a universe to the interval \([0, 1]\) such that the sum of all the mappings is equal to 1. To define a probability distribution, we use the expression

$$\{ \text{value}_1 \rightarrow \text{value}_1, \text{value}_2 \rightarrow \text{value}_2, \ldots, \text{value}_n \rightarrow \text{value}_n \}$$

where \(\text{value}_1\) to \(\text{value}_n\) are elements of a set, and \(\text{value}_1\) to \(\text{value}_n\) are the associated probability values, which together must add up to 1.

BSL then changes the grammar rule of the choose rule from

```
choose id, in id, with guard do Rule, if rule, end choose
```

(2)

to

```
choose id, in id, with guard using dist do Rule, if rule, end choose
```

(3)

We provide a small set of constructions for some well-known probability distributions. The uniform distribution function

```
uniformDistribution(value)
```

(4)

defines a uniform distribution over the finite set \(\text{value}\); e.g., we can define a uniform distribution for the numbers 0 to 100 with the expression

```
uniformDistribution(toSet([0..100]))
```

The Poisson distribution function

```
poissonDistribution(value)
```

(5)

defines a Poisson distribution with expected \(\text{INTEGER value value}\).

3 Scheduling

Scheduling allows us to artificially control the passage of time for the different agents in multi-agent environments. Assuming zero-time execution for ASMs, it is safe to assume that, at every point in a concurrent ASM environment, there is a subset \(S\) of the set of \(\text{Agents}\) executing. We call \(S\) the set of scheduled agents.

Although CoreASM offers the primitives \(\text{suspend}\) and \(\text{resume}\) to control how agents run in concurrent environments, those primitives interact with CoreASM’s internal scheduling policy. Consequently, even if an agent is “resumed”, there is no guarantee that it will run the next step, because it may be discarded by CoreASM’s scheduling policy. To address this problem, BSL changes the semantic of the CoreASM’s scheduler, and it allows us to define how agents are scheduled by the engine directly inside BSL scripts; more precisely, instead of
suspended and resuming agents, we schedule them using a user-defined scheduling policy (not CoreASM’s scheduling policy).

To force an agent to run at the current step, we use the schedule policy

\[ \text{schedule } \text{id} \] (6)

When scheduled, the agent executes the rule that corresponds to its program.

Now, to provide rich syntax for the definition of scheduling policies, we adapt the grammar rules for ASM rules [1] to scheduling policies. The no-op policy is

\[ \text{skip} \] The block policy

\[ \text{par } \text{Policy}_1, \text{Policy}_2, \ldots, \text{Policy}_n \] (7)

evaluates the given policies in parallel. The choose policy

\[ \text{choose } \text{id}_1 \text{ in } \text{id}_i \text{ with guard } \text{using dist} \text{ do } \text{Policy}_1 \text{ ifnone } \text{Policy}_2 \text{ endchoose} \] (8)

chooses according to the distribution from the finite set \( \text{id}_i \) an element that satisfies the guard, assigns it to \( \text{id}_1 \), and evaluates \( \text{Policy}_1 \). If no such element exists, then \( \text{Policy}_2 \) is evaluated instead of \( \text{Policy}_1 \). The conditional policy

\[ \text{if guard then } \text{Policy}_1 \text{ else } \text{Policy}_2 \] (9)

if guard holds, then it evaluates the \( \text{Policy}_1 \); otherwise, it evaluates the \( \text{Policy}_2 \). The forall policy

\[ \text{forall } \text{id}_1 \text{ in } \text{id}_i \text{ with guard } \text{do } \text{Policy} \text{ endforall} \] (10)

takes from the (finite) universe \( \text{id}_i \) all elements that satisfy the guard, and assigns them one by one to \( \text{id}_1 \), to evaluate the policy. The set of scheduled agents is the aggregation of the evaluation of the policy using all the different values of \( \text{id}_i \). The let policy

\[ \text{let } \text{id}_1 \text{ = value}_1, \text{id}_2 \text{ = value}_2, \ldots, \text{id}_n \text{ = value}_n \text{ in } \text{Policy} \] (11)

assigns to the (logical) variables \( \text{id}_1 \) to \( \text{id}_n \) the values \( \text{value}_1 \) to \( \text{value}_n \), respectively; then, it evaluates the \( \text{Policy} \). The case policy form

\[ \text{case value of } \text{value}_1 : \text{Policy}_1, \ldots, \text{value}_n : \text{Policy}_n \text{ endcase} \] (12)

evaluates the policy \( \text{Policy}_i \) if and only if \( \text{value} \) is equal to \( \text{value}_i \). If two values \( \text{value}_1 \) and \( \text{value}_n \) are equal to \( \text{value} \), then both \( \text{Policy}_1 \) and \( \text{Policy}_n \) are evaluated.
4 Concluding Remarks

While there is still a long road ahead to be able to provide biologists with the tools they need to adequately model biological systems using software engineering tools, BSL takes us one step closer to that goal. Though we only offer a couple of new extensions, BSL is able to grow as biologists find the need for new language primitives to express their models naturally.

Having BSL specifications of biological systems will allow computer scientists to test and understand the models hands-on. This common understanding of a biological system promotes collaboration between biologists and computer scientists with the objective of discovering the necessary components and conditions that promote a particular biological behaviour. Once the causes and conditions are identified, we can use the ASM notions of abstraction and refinement (see [1]) to abstract the biological models, and then refine them as software systems that exhibit the desired biological behaviour, contributing to an era of truly biologically-inspired software.

References

LIFE: Load Balancing Inspired by Filament Structures

by Daniel Schreckling, Eric Rothstein Morris and Chrystopher L. Nehaniv

LIFE: Load Balancing Inspired by Filament Structures

Daniel Schreckling¹, Eric Rothstein², and Chrystopher Nehaniv²

¹ IT-Security Group, Institute of IT-Security and Security Law
University of Passau, Passau, Germany
² Royal Society Wolfson Biocomputation Research Laboratory
University of Hertfordshire
Hatfield AL10 9AB, UK

Abstract. We introduce LIFE: A proof-of-concept implementation which changes a computational structure to achieve load balancing properties. Changes are triggered by the environment, e.g., specific request, and follow a pattern observable in the growth of filamentous multicellular organisms. The resulting ring structure dynamically and autonomously adapts to different request leads by allocating and releasing resources. This behavior is achieved without global knowledge and only through local interactions.

Keywords: Filament, Load Balancing, Multicellular Growth, Simulation

1 Introduction

Filaments are ubiquitous structures in Biology. They appear to follow simple growth and replication rules. Despite this simplicity their changes in structure and functional specializations appear to support the survival of complete organisms, e.g., by producing sufficient nutrition for the rest of the organism or the structure to stand and grow. Simple multicellular organisms with differentiation of cells into a few different types provide natural examples of adaptive and robust division of labour in dynamic environments.

These structures are particularly interesting in the context of the work conducted in the European project BIOMICS - Biological and Mathematical Basis of Interaction Computing - as they use simple and local interactions to detect specific conditions to adapt their overall behavior. Simple behavioral cell transformations can change structure and functionality of the system under consideration to achieve the ultimate goal: survival. As these transformations also change observable behavior and thus functionality of the organisms they can be considered simple instances of interaction computing. Hence, we consider filaments as a feasible and useful direction for study.

The goal of this contribution is to apply the principles we can observe in the filaments domain in the computer science domain. By applying simple interaction mechanisms we will try to build computer science systems which react to their
environment to maintain a specific quality of service but remain stable and alive by using the available resources in the best way feasible. More specifically, instead of distributing vital biochemical substances in an organism we will exploit the interactions observed in filament to distribute the current system state and only act upon detection of specific states by simple cell transformations.

This work aims for a simple prototype as a case study for the interaction computing machine (ICM) model developed in BIOMICS. We study its feasibility, power, and limitations to guide the implementation of the execution framework. Despite this preliminary state, we will try to reuse several components of the results of this contribution.

2 Background

To better understand our mapping from filaments to computer science, this section focuses on the specific systems which inspired our work.

2.1 Filamentous Growth

Filamentous growth can be observed in numerous biological systems. Our work was specifically inspired by *Anabaena*. *Anabaena* is a cyanobacterium which grows in multicellular filaments and exists as plankton. Through its ability to produce neurotoxins and work as a natural fertilizer, it can also live in symbiotic relationships with plants. *Anabaena* has the ability to fix nitrogen in specialized cells called heterocysts. These cells differentiate from regular cells if specific conditions are determined in the environment. Such a specialized cell in the multicellular filament is also supported by regular vegetative cells surrounding it. The latter deliver carbohydrates, e.g. consumed from a plant, to heterocysts and obtain nitrogen in return.

This organization in to cell types comprises a basic and effective division of tasks among the different types of cells in the filament.

Due to its simplicity, *Anabaena* has been subject to models which try to reconstruct the principal structures of multicellular filaments of *Anabaena*. Most popular are the grammars defined in the models by Lindenmayer [9].

2.2 Load Balancing

In general, load balancing tries to distribute some system workload among available resources and tries to optimize resource usage as well improve the quality of the services inducing the workload.

In computer science, we mainly distinguish static and dynamic load balancer. Static load balancer try to distribute load uniformly among all resources available. While this distribution can be computed and decided upon easily, it often shows poor resource usage and thus also lacks appropriate service quality.

In contrast, dynamic load balancers try to allocate resources to computational tasks in such a way that the overall system load remains minimal. Of course, this requires awareness of the load in each single system. As a consequence, additional communication overhead is generated when accumulating the overall system view and determining the best resource choice.

Defining good dynamic balancers is a complex task which has produced a huge body of work in the last decades. The complexity of the design of good load balancers is nicely illustrated by Azar et al. in [1]. The authors explain how the power of two choices can have a tremendous impact on the maximum load a system can deal with. They show that by just giving a load balancer two options instead of one, it could decrease overall system load exponentially. Various systems have been investigated focusing on how they can account for this property [4].

In this early study, we do not aim for an optimal load balancer. Instead we want to deliver a proof-of-concept implementation. However, we are also aware of recent results [5] which show that the so-called join-the-shortest queue (JSQ) strategy does not necessarily outperform the join-the-idle queue (JIQ) strategy. Even more interestingly, the work by Mukherjee et al. shows that in many-servers heavy-traffic scenarios the JIQ strategy is a strategy which is close to an optimal solution. These results and the simplicity of the JIQ policies motivated the use of JIQ in our proof-of-concept.

3 Load Balancing and Filamentous Growth

As explained above, the structural growth of filaments and specialization is controlled by rather simple mechanisms and can be described using simple grammars. We are going to exploit this simplicity and map it into a framework which is able to dynamically distribute workload over a set of available resources.

3.1 Approach Overview

In our model, we abstract from the biological behavior of filaments as described in our introduction and adapt it such that its natural simplicity is maintained but that it can also serve the more complex task of load balancing. To understand our adjustments, we first introduce the main idea of our approach.

Filament behavior is mostly controlled by the environment. Local and environmental conditions may influence local behavior and impact the overall filament structure. Through these decisions, filaments achieve global behavior beneficial for them and possibly for their environment or their symbiotic relationships. Load balancers also have to take decisions based on local as well as global conditions in order to efficiently use the resources available for their task. This observation motivates the use of simple structural modifications exhibited by filament to balance load over resources. Our analogy maps molecules and light used by filaments into requests which arrive in our system. The cells of filaments are computational entities in our system which are living in a reactor.
This reactor offers a set of resources which can be used by the computational entities. Their efficient use and the appropriate serving of requests is the main goal of the system.

3.2 Model Adjustments

Of course, this mapping from biology to computer science requires some adjustments of the original filament model. The first adjustment modifies the filament structure. Instead of growing tree-like structures, LIFE will generate rings of cells. Main advantage of this structural modification is the fact that it can grow and shrink as needed. Linear or tree structures of filaments simply keep on growing. As they do not implement some load balancing mechanism they do not have to free resources. Thus, filament structures do not shrink. In contrast, our structure need to shrink as a shrinking structure corresponds to freeing resources. Further, distribution of workload inside a ring is easier than in linear or tree structures. Finally, the difficulty to infer global knowledge in such structures additionally motivates the use of ring structures.

Shrinking of structures also requires an additional adjustment of the dynamic behavior of filaments: Cell death. Under certain conditions LIFE allows cell death. Cells die to release resources. However, death is a structure preserving operation, i.e. a dying cell will not destroy the ring but only decrease its size.

![Diagram](image)

**Fig. 1. Schematic illustration of LIFE components and their interaction**

Finally, our model maps the specialization of cells to the generation of new ring structures, i.e. instead of specializing a new cell into a cell storing nitrogen to support its neighboring cells, the cell specialization will generate a new ring. We motivate this adaptation by mapping each structure to a pool of threads or processes. The structures can use these resources. If it reaches its saturation-size, i.e. each cell is mapped to a resource, it can generate a new ring which has not reached the saturation-size and is available to serve new requests.
3.3 Components

Figure 1 depicts the components used to model our load balancing system.

**Reactor** A reactor is a container which hosts cells and substances (see below). It can further receive requests from its environment, e.g. a user, some web service, a device, etc. The reactor selects cells and forwards them the requests it has received. This selection process follows simple rules.

The reactor has a set of (possibly distributed) resources at its disposal which can be used to answer requests entering the system.

**Requests** The system input are requests which are generated by the environment. These input can be any computational task, starting from simple tasks, computing a route, or simply accessing resources such as in a web server. Requests also keep the system alive. Without a request the system will die.

**Substances** Substances can be considered as global signals in the system. They are global quantities which can be observed or consumed by any entity. When special substances reach a specific concentration, they can trigger actions.

**Cells and Rings** Cells are computational entities which all run the same program. They have one left and one right neighbor. Such neighbors are also cells. A cell which has itself as left and right neighbor is also called a ring of size one.

Cells only have local knowledge, i.e. they only know their neighbors and the knowledge they are able to accumulate through control messages, called **signals**.

All cells only live for a specific time which is determined when the cell is born. At that event the cell directly starts dying, i.e. a timer counts to zero. If the timer reaches zero the cell dies. Before dying, a cell ensures that its death does not destroy the ring structure, i.e. it ensures that its neighbors link to each other. Hence, if the cell is a ring of size one, the ring dies with the cell.

During their lifetime, cells can serve requests if they are not busy. If they serve a request, cells are busy. Busy cells are not able to die, i.e. the death timer is stopped. After serving a request, the cell goes into idle mode and resets and restarts its death timer.

If required, cells can also divide into a parent cell (the cell itself) and one offspring. During division, offspring are generated inside a ring. In contrast, a cell can also duplicate itself. The cell which duplicates continues to exist but generates a new offspring outside the ring, i.e. duplication generates a new ring. Both, division as well as duplication can be triggered by local conditions and substances (see above). The latter are generated by cells and their actions.

Finally, cells can receive, forward, and send **signals** to their left neighbors.

**Signals** Signals are simple control messages which propagate inside a ring in one direction. Each signal possesses a hop count which represents the number of cells that have processed it, store their origins and can accommodate a request.
3.4 Load Balancing in Action

All requests arriving at our system are first distributed by the reactor. For each incoming request, the reactor chooses a cell among all existing cells in the reactor. We currently choose all cells according to a uniform distribution. The reactor generates a new signal and sends it to the selected cell. This signal has hop count one, marks the selected cell as its origin and accommodates the request.

As soon as the cell recognizes signals in its inbox, it will inspect them. If the signal contains a request, the cell will check its current mode. If the cell is in idle mode, it can serve a request. Thus, it extracts the request from the signal and starts serving the request, i.e. it uses the resource assigned to perform the computational task required by the request. During this task, the cell switches its mode to busy. In order to survive the task, the cell also stops the death timer. On completion, the cell resets, goes into idle mode, and restarts its death timer, i.e. each request refreshes the cell and extends its time to death.

If a cell is in busy mode and receives another request inside a signal, it cannot serve this request. Therefore, it must forward the request to its left neighbor. For this purpose, the cell first extracts the origin of the signal and checks whether it originated at the left neighbor. If this is not the case, the cell updates the hop count of the original signal and sends it to the left neighbor by inserting it into the inbox of this neighbor. Additionally, if the new hop count exceeds a specific threshold, in our case, the saturation-size of the ring, the cell also emits an element of substance NEWRING.

In case the left neighbor is the origin of the signal, the current cell knows that the signal traveled around the ring without finding a cell which could serve its request. As a consequence, the cell must take action and prepare a new resource to serve the request. To achieve this, it has two options: increasing the ring size by division or generating a new ring by duplication. The substance NEWRING decides on which option to take. Please note that the pure existence of the substance NEWRING, does not imply that all cells in a ring are busy. During the propagation of a signal inside a ring, a cell may be in mode busy when a signal arrives at the cell, but it may become idle directly after the signal has passed it. Thus, the substance NEWRING is only an indicator for the time it needs to find another idle cell in one ring. As a consequence, the cell only performs duplication when NEWRING exceeds a pre-defined duplication threshold.

As described above, duplication generates a new cell outside the ring and maintains its original parent cell. After duplication the parent cell puts a new signal into the inbox of the offspring. This signal has hop count zero, marks the new offspring as its origin and contains the request which arrived at the parent cell. Hence, the request is now served in a new ring. During duplication, the cell consumes the amount of saturation-size of the substance NEWRING.

In case the substance NEWRING has not exceeded duplication threshold, the cell performs a division. The offspring becomes the new left neighbor of the cell where the signal arrived and the right neighbor of the origin of the signal. After division is completed, the signal is again sent to the new left neighbor. As this neighbor is in idle mode, it can directly serve the request.
4 Simulation Framework

We implement the main aspects of LIFE as a proof-of-concept. This implies, that we do not serve real requests. Instead, they are simple text messages injected into the system. Our main focus is on the interaction between cells and rings and the overall behavior of LIFE. In the remainder of this section, we explain the main architectural features of our prototype and show first preliminary results.

The architecture of the LIFE prototype (see also Figure 2) is based on node.js\(^1\), an event-driven, non-blocking JavaScript runtime which is based on Chrome’s V8 JavaScript engine. For visualization of cells and rings and for drawing quantity graphs, we use vis.js\(^2\), a browser based visualization library.

![Diagram of LIFE prototype](image)

**Fig. 2.** High-level architecture of the LIFE prototype

Our prototype architecture follows the model-view-controller software design pattern. The model and parts of the controller run inside node.js on a server. View and controller run inside a browser the user can use to control the simulation and view the result of it.

The model of our system comprises two main JavaScript modules, the Reactor and the Cell class. The reactor implements a server by using the websocket library see. It listens for incoming connections of client-controllers and client-views, the browser side visualization. The reactor hosts a set of cell instances and allows the selection, creation, and deletion of cell instances. Each generated cell

\(^1\) [http://nodejs.org/](http://nodejs.org/)
\(^2\) [http://visjs.org/](http://visjs.org/)
in the model also stores a reference to the reactor class. In this way, each cell 
can communicate its actions to the reactor. The latter will update the view, i.e. 
all connected clients, accordingly. The same holds for the controller. The reac-
tor offers appropriate primitives to, e.g., starting the simulation, resetting it, or 
creating a new cell in the reactor. As soon as a client manipulates the simu-
lation by using its user interface, the corresponding commands are sent to the 
reactor which takes the correct action. Further, the reactor class also artificially 
generates requests to the system.

Each cell in our system inherits from the EventEmitter class and its im-
plementation follows the asynchronous idiomatic, i.e. all actions in a cell are 
implemented non-blocking, event-driven, and thus asynchronous fashion. Hence, 
we can consider cells to execute concurrently. The model is directly reflected 
as described above, i.e. a cell possesses references to left and right neighbor, a 
queue for incoming signals, a death timer, and primitives for cell division and 
duplication. Further, it also stores a reference to the reactor, to also publish all 
its actions to the reactor that hosts this cell.

On the client side, the view exists of two main classes: Cells and ReactorConnection. 
While the latter manages the connection of the client to the reactor, the class 
Cells uses the vis.js library to visualize the state of all cells. For this purpose, 
it abstracts from the internals of the Cell class and simply stores the current 
connectivity of a Cell. Updates from the reactor are received via the Reactor- 
Connection and update the view accordingly. The same connection is used in 
the Control class to send control information to the reactor.

5 Future Work

Currently, our work only considers circular structures. They simplify the rules of 
interaction, growth, and degradation. To better align with the models of filament 
growth, we also think about introducing tree structures. They are particularly 
interesting as information derivation becomes more complicated. In our model, 
the estimation of the structure size is strongly dependent on its shape. In case 
of tree structures, we think that it may become necessary to also generate spe-
cialized cells which are responsible for particular factions of the structure.

Another interesting modification of our model we will investigate in the fu-
ture concerns the repair functionality of a cell. Our cell model currently repairs 
the ring before a cell dies. We may simplify the cell behavior and simply ig-
gnore the invariants which must be maintained for the structure the cell lives in. 
This implies, that we would destroy such structure and generate new fragments. 
Whether we force them to reconstruct in their original structure and expressing 
some potential to do so through specifying additional rules or let them reorganize 
in their own more complex structures will be subject to further experiments.

While this model appears trivial, we are convinced that simple rules can be 
used to build more complex software. With our model, we are able to mimic 
properties of complex self-balancing mechanisms. However, we may adjust our 
mappings from nature to computer science and associate preferred functionality
with rings hosting different cell types which have different sizes. Interaction between such rings or structures of different shape may produce new more complex functionality. We will even try to form hierarchies of structures where structures of structures, e.g., rings of rings, may be considered. In this regard, we envision to also include additional insights from biology. Here, we are particularly interested in the work of Lam and Silvestre [2] who consider the interaction between different types of Anabaena filaments.

LIFE is experimental work which uses the combination of load balancing and the structural growth of filaments as a study subject. While the results of this implementation is interesting in itself, we also focus in the re-usability of our architecture for similar problems derived from biology. For this reason, we will also consider introducing BSL into our underlying simulation framework [8,9]. We are convinced that this will simplify the specification of similarly biologically inspired systems. Specifications will boil down to simple behavioral descriptions. As this language also supports hierarchical structures, the previously described future extensions will also be simplified by BSL. Finally, as the Interaction Computing Machine [7,6] guided the design of BSL, we can also exploit its features to define the interaction between cells and even the more complex ring structures by using interaction policies. The same language constructs can be exploited to specify the selection rules for cells which must receive requests. These simple rules will allow for more sophisticated selections.

6 Conclusions

We developed a mechanism which was inspired by simple interaction rules between biological cells to achieve system growth in symbiotic relationships. This inspiration led to self-stabilizing system which reacts to environmental input. While we are not sure yet that our mechanism is able to actually provide a good model for a load balancing system, our proof-of-concept implementation supports the interaction computing model developed in BIOMICS: Simple local interactions which are triggered and driven by inputs from the environment can achieve rather complex and self-maintaining behavior.

While our current proof-of-concept implementation is tailored to our specific use case, load balancing, we see good opportunities to reuse our implementation and enhance it with appropriate components which allow a complete realization of the interaction computing model [6] including the use of BSL as its specification language. The MVC model, we used for the implementation of this prototype will support this elaboration. In fact, some deficiencies of the current implementation reveal the power of the ICM model and BSL: Extending our simulation to several collaborative rings which can serve different functionalities would require a mechanism which can control the interaction between different rings, based on different environmental conditions. The machinery of the ICM and the power of BSL simplify this task tremendously, as simple interaction rules can be defined based on the different properties of the cells involved.
Acknowledgments

The research by the authors leading to these results was funded in part by the European Union’s Seventh Framework Programme (FP7/2007-2013) under the BIOMICS project, grant agreement no. 318202. We would also like to thank Egon Börger for his very helpful feedback on first versions of our protocol. It helped to make our model more concise.

References