A structural perspective on the dynamics of biochemical systems

Sylvain Soliman

HABILITATION À DIRIGER DES RECHERCHES

Spécialité : Informatique

par

Sylvain SOLIMAN

A structural perspective on the dynamics of biochemical systems

Présentée le 7 décembre 2016 devant le jury composé de :

M.	Vincent DANOS	(Rapporteur)
M.	Simon DE GIVRY	(Rapporteur)
M.	Nicolas LE NOVÈRE	(Rapporteur)
M.	Philippe DAGUE	
M.	Alain DENISE	
M.	François FAGES	
M.	Christophe SOULÉ	

H.D.R. préparée au sein de l'EPI LIFEWARE

CENTRE DE RECHERCHE
Inria Saclay – Île-de-France 1 rue Honoré d'Estienne d'Orves
Bâtiment Alan Turing
Campus de l'École Polytechnique
91120 Palaiseau

Abstract

In recent years Systems Biology has become a rich field of study, trying to encompass all the information that has become available thanks to the new high-throughput techniques of biologists.

Fifteen years ago, a fundamental breakthrough was the publication of Kurt Kohn's map of the cell cycle control in mammals. Its similarity with electronic circuits was crucial in both making it impossible for humans to comprehend fully, and in prompting the use of formal methods.

Since then however, the networks built by biologists and modellers have continued growing bigger, filled with more and more mechanistic details, especially recently acquired post-transcriptional information, but lacking most of precise kinetic data. Because analysis techniques providing dynamical insights mostly rely on complete kinetic information, what was challenging for the human ten years ago is now a challenge even for computers,

In this manuscript we will try to give an account of our work of the last twelve years, centered around the question of model. We will define more precisely what this object is, formally, and try to handle the challenges raised by the ever growing amount of data, and corresponding size of models developed in Systems Biology. Our main focus will be the links between the structure and dynamics of those models, seen as means to use several formal methods, like Constraint Programming, to reason on their dynamics. Because of the size issue we will also discuss the question of model reduction, and the related relationships between models, formalisms and interpretations.

The discrete nature of the structure underlying a model might seem opposed to the continuous dynamics often associated via differential equations to that model. Nevertheless, we hope to demonstrate that the gap between those two views is quite artificial, and that recent results offer very promising perspectives to bridge it.

Keywords : Systems biology, Petri nets, Constraint Programming, Abstract Interpretation, Model-Checking.

Une perspective structurelle sur la dynamique des systèmes biochimiques

Résumé

La Biologie des Systèmes est depuis peu devenue un domaine de recherche florissant qui tente de tirer parti de toutes les informations que les techniques à haut débit des biologistes ont rendues disponibles.

Il y a une quinzaine d'années, la publication de la carte du contrôle du cycle cellulaire des mammifères par Kurt Kohn a été une avancée fondamentale. Sa similarité avec un circuit électronique est cruciale dans l'impossibilité pour des humains de l'appréhender complètement mais aussi dans l'idée d'utiliser pour cela des méthodes formelles.

Cependant, depuis cette carte, les réseaux construits par les biologistes et modélisateurs ont continué à croître, enrichis de détails mécanistes de plus en plus précis, en particulier des informations post-transcriptionnelles acquises récemment, mais sans données cinétiques précises. Or les méthodes d'analyse qui fournissent des informations dynamiques s'appuient principalement sur une information cinétique complète, par conséquent, ce qui était un défi pour l'humain il y a dix ans l'est devenu pour l'ordinateur aussi.

Dans ce manuscrit nous allons tenter de donner une idée de notre travail des douze dernières années, centré autour de la notion de modèle. Nous allons définir plus précisément, et formellement, cet objet, et essayer de répondre aux défis liés au foisonnement des données et à la croissance des modèles de Biologie des Systèmes. Notre approche est fondée sur les liens entre structure et dynamique d'un tel modèle, ces liens permettant d'utiliser de nombreuses

méthodes formelles, comme par exemple la programmation par contraintes, pour raisonner sur la dynamique de ces objets. Le problème de la taille nous amènera par ailleurs à envisager les questions de réduction de modèle, mais aussi de liens entre modèles, formalismes et interprétations.

La nature discrète de la structure qui sous-tend un modèle peut sembler opposée à la continuité de la dynamique qui lui est souvent associée via des équations différentielles. Cependant nous espérons démontrer que cette séparation est essentiellement artificielle, et que les résultats récents offrent des perspectives prometteuses de la réduire.

Mots-clefs : Biologie des systèmes, réseaux de Petri, Programmation par Contraintes, Interprétation Abstraite, Model-Checking.

As an adolescent I aspired to lasting fame, I craved factual certainty, and I thirsted for a meaningful vision of human life - so I became a scientist. This is like becoming an archbishop so you can meet girls.

M. Cartmill

Remerciements

Je tiens d'abord à remercier l'équipe Lifeware, et même l'ancienne équipe Contraintes, dont les membres ont depuis des années contribué à mon travail, que ce soit à travers des publications communes, des séminaires de groupe ou même des discussions à table ou en pause café. L'ambiance agréable dont je profite n'est pas étrangère au plaisir renouvelé que j'ai d'aller travailler le matin. Même les anciens membres sont pour certains devenus des amis, ce qui montre leur incroyable capacité à supporter mon humour, mes références musicales des années 80 et ma manière de parler trop fort. Merci en particulier à François qui a su accepter mes particularités (comment ça je n'aime pas voyager ?) tout en ne cessant de me pousser dans la direction la meilleure pour moi.

Je remercie aussi ma famille, notamment mes parents qui sans bien savoir ce qu'est une HDR n'en ont pas moins été encourageants depuis bien longtemps. Mes amis hors du monde académique qui ne comprennent rien à mon travail mais continuent à feindre l'admiration, dans l'espoir secret que je pourrai prochainement réparer leur imprimante défectueuse. Enfin mon épouse, Lisa, qui, malgré des moments difficiles, n'a jamais varié dans son soutien inébranlable ; c'est à elle que je dédie ce mémoire.

Contents

Research is the process of going up alleys to see if they are blind.

Marston Bates

Chapter 1

Introduction

Summary

I would first like to point out that most of the work presented in this thesis has already been published. Furthermore, it is usually common work with several of my colleagues, François Fages my former Ph.D. advisor and leader of the team, but also Ph.D. students and interns that spent some time in the LIFEWARE team and also researchers from other institutions.

I thus decided to compile these results, not as a compendium of novel research, or as a proof of the amount of work I did *by myself*, but rather with the aim to share a perspective upon this emerging body of work. I intend to shed a new light on some of these articles and to show how all the issues they tackle happened to be on a single path, mine.

Indeed, I will journey through different research areas, all related to the quite fuzzily defined field of Systems Biology, but that might seem very different at first sight. Notably:

- Model-Checking;
- Constraint Programming;
- Petri-Nets structural analysis;

- dynamical systems' steady-state analysis, S-Systems and Chemical Reaction Network Theory;
- Abstract Interpretation and typing;
- graph morphisms;
- Ordinary Differential Equations.

The message that I would like to convey is that the field of Systems Biology, through its constant feed of knowledge and data, is always at the edge of the *qualitative* and *quantitative* worlds. The models, usually based in the qualitative world, if only by their original nature: a drawing, never cease to grow bigger and bigger, encompassing more and more knowledge. On the other hand, the hope of biologists is to match what has recently become a huge amount of quantitative timed data to those models.

These purposes seem quite different: the more qualitative information you include, the more quantitative measurements are necessary to fit your model, the race never ends. However to reconcile them, one can take insight in the advice of the computer scientist that a model, whether qualitative or quantitative, should always be the smallest that can answer the question at hand.

This idea, though it might seem contradictory with both the knowledge-storing modeler and the data-greedy biologist, leads to an interesting remark: when a full quantitative model is out of reach, e.g., through lack of precise kinetic data for some reactions or compounds, its structure might be detailed enough to provide the required information about its dynamical behavior, validating (or not) the experimental data.

All the work I present here, all the use I make of the techniques mentioned above, is guided by this idea: extracting from the rich structure of the big models key informations allowing the modeller to reason on their dynamical behavior. In other words **relating the structure of biochemical models and their dynamics**.

1.1 Context – Systems Biology and Modelling

Wikipedia defines Systems Biology as follows:

> Systems biology (also known as Systeomics) is the computational and mathematical modeling of complex biological systems. An emerging engineering approach applied to biomedical and biological scientific research. Systems biology is a biology-based inter-disciplinary field of study that focuses on complex interactions within biological systems, using a holistic approach (holism instead of the more traditional reductionism) to biological and biomedical research.

The focus is therefore on an integrative approach, and the tool chosen to realize this integration is the *model* that represents the system under study, as opposed to the single species (protein, molecule, etc.) that was the classical object of interest.

1.1.1 What is a model?

The precise definition of what a model really is has eluded me since my focus shifted from the well-defined world of logics to that of Systems Biology about ten years ago. Does changing the value of a given parameter or initial value in a system of ODEs representing some biochemical network make it a new model? Does changing the kinetic law of some enzymatic reaction from Michaelian to Hill create a new model? Does the quasi-steady-state approximation and the corresponding simplification change the model? What if only the names given to compounds change?

Working with biologists and modellers involves learning that there is no clear definition of what a model is, where it starts, nor where it ends. Since the (Wikipedia) definition of Systems Biology involves some computational and mathematical approaches however, there is, each time, the need to formalize a model, to decide what is part of it and what is not.

We were lucky to be able to rely on the development of the Systems Biology Markup Language (SBML) [42] as a Nature Publishing Group endorsed standard for models. Nevertheless, especially in its recent Level 3 incarnation, it has become clear that it tries to encompass many kinds of models and to avoid making choices that would exclude anything. SBML can now encode both logical models *à la* Thomas, and flux-based models for balance analysis, on top of state-transition systems, reaction-based continuous and stochastic models, etc.

1.1.2 Simple Biocham Modelling Language

In this manuscript, we will focus on the subset of SBML that was at its origin, i.e., reaction systems. A model will therefore be, at its core, a set of reactions between biochemical species.

From the SBML Frequently Asked Questions:

"A common abstraction used when describing cellular phenomena is to describe the system as a set of chemical entities linked by processes (reactions) that can transform one entity into another or transport entities between compartments."

This focus on reactions is shared by the BIOCHemical Abstract Machine (BIOCHAM), our modelling software [2, 26]. Though another crucial component of a BIOCHAM model is the specification of its behavior, this thesis will assume that by model we intend to describe the reaction system, and therefore that core-SBML and core-BIOCHAM are mostly interchangeable as formats to describe a model. Section 1.2 will give more details about the formalisms used, including SBML, BIOCHAM but also bipartite graphs and Petri nets.

Note that we will also avoid adding powerful constructs like events, though they exist both in SBML and BIOCHAM. Indeed, they add a huge expressive power and allow a direct representation of hybrid systems [3] like those resulting from some of the methods we will expose (e.g., see Section 3.3), but this is usually at the expense of a gap between the structure and the intended semantics of the model.

1.1.3 A picture is worth one thousand words

Through our collaborations with biologists from INSERM, CNRS or INRA, as well as from our reading of the modelling articles in biology, we have observed that modellers rely often on a drawing to represent, informally, their knowledge about the system they are about to model. From big maps like that of Kohn [46] (see Figure 1.2) to simple graphs like the six variable cell-division model of John Tyson [62] displayed in Figure 1.1, everything starts with a drawing.

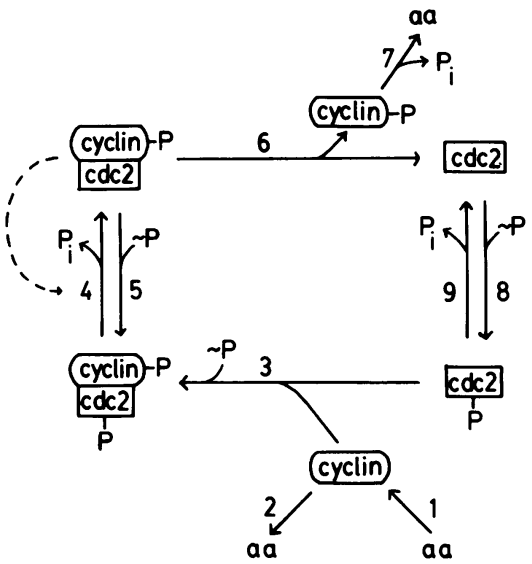


Figure 1.1: Figure 1 of [62], typical of the way many modellers describe the system they are studying.

Quite naturally, when the question of computational tools arise, the tendency is to go towards nice-looking tools offering little if any analytical feature at all, like CellDesigner (`www.celldesigner.org`), or tools that can interoperate with it but mostly rely on basic graph-theoretic concepts like Cytoscape [58] and its BiNoM plugin [64].

Indeed, the formalization of such a drawing can be captured as a model with the meaning given above, but as a *structural* model, i.e., a model without any kinetic information. Unfortunately, one might conclude that then, the available formal/computational analyses are restricted to notions like simple paths and first or second order neighbor to have an idea of what is happening in the model. When biologists do not have the full kinetic data, they give up completely on any dynamical analysis of their network.

This is definitely not satisfactory and the works presented in this thesis, notably in Chapter 2, show that it is not a necessity either since the structure of the model already contains a lot of information about its possible dynamics.

1.1.4 Size is a growing issue

Fifteen years ago, a fundamental breakthrough was the publication of Kurt Kohn's map of the cell cycle control in mammals [46]. Its similarity with electronic circuits (see for instance Figure 1.2) was crucial in both making it impossible for humans to comprehend fully, since it represented about 500 different compounds involved in about 800 reactions, and in prompting the use of formal methods [8].

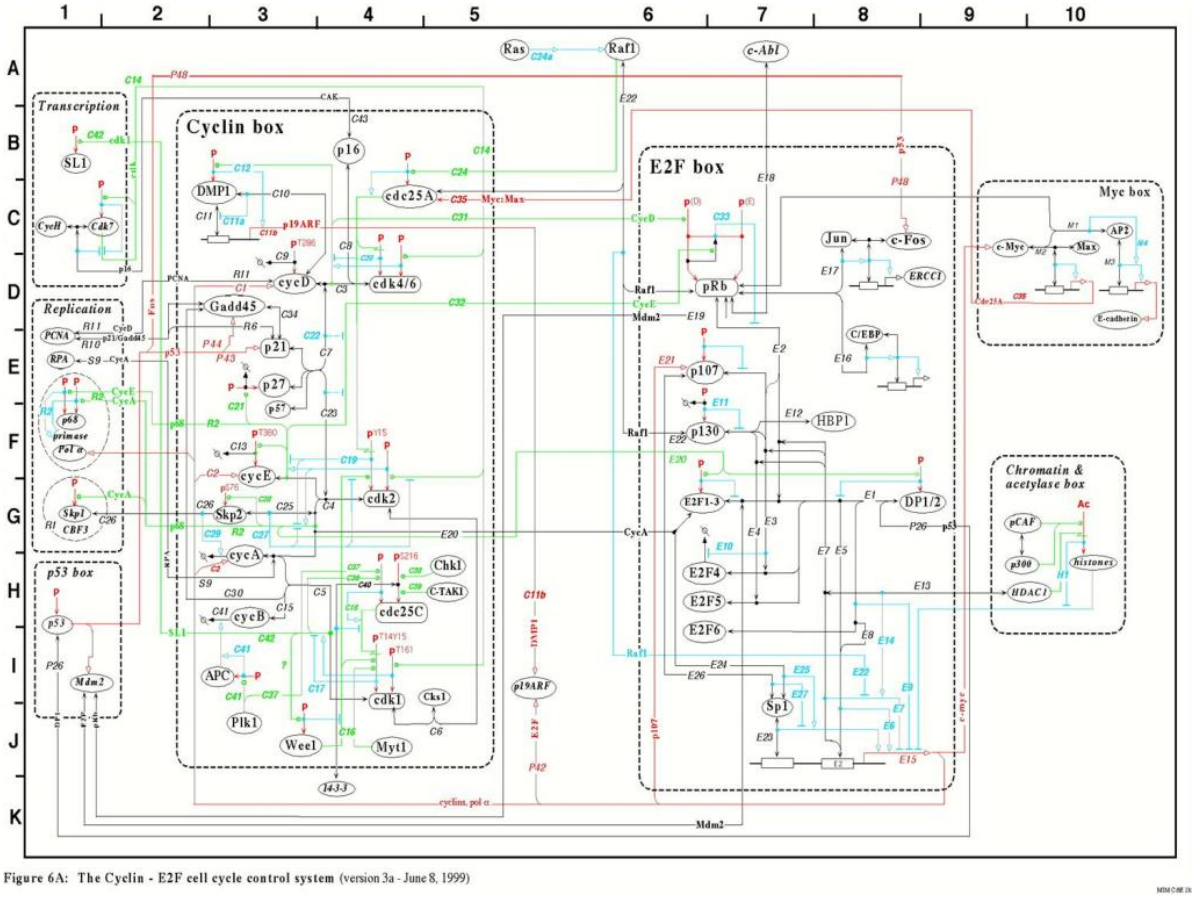


Figure 6A: The Cyclin - E2F cell cycle control system (version 3a - June 8, 1999)

Figure 1.2: A part of Kurt Kohn's map of the mammalian cell cycle control [46]

Since then however, the models built by biologists and modellers have continued growing bigger and bigger, filled with more and more mechanistic details, especially recently acquired post-transcriptional information, but lacking most of precise kinetic data. For instance, the *average* BioModels' entry now[1] contains more than 450 compounds and about the same number of reactions, more than tenfold what it was in 2005 when BioModels was started. The biggest models have thousands of species and reactions, see Section 3.2 for some examples. The extreme has become the norm, and therefore, what was challenging for the human ten years ago is now a challenge even for computers: analysis tools either fail to scale to *big* models with hundreds of reactions and compounds, or fail to provide any dynamical insight when full kinetic information is not available. This defines our main challenge: to take into account the size and

[1] As of September 16th 2014, the 28th release of the BioModels database (see `http://www.ebi.ac.uk/biomodels-main/static-pages.do?page=release_20140916`)

lack of information of current biochemical models and nevertheless bring forward some analyses of their dynamical behavior. This will also prompt our use of the structure of the models, not only for Model-Checking, as was originally done in BIOCHAM [25, 26], but also for reducing models to a more manageable size using Constraint Programming in Chapter 3.

1.1.5 From qualitative to quantitative

Another common trend in Systems Biology is the multiplication of tools and formalisms to analyse models. This is both very positive on the methodological side and a little bit puzzling for the unexperienced modeller.

In Chapter 2 we present recent results on necessary conditions for the multistationarity of a system, i.e., its ability to present homeostasis. The different steady states might for instance correspond to the possible outcomes of cell differentiation. These results already rely on some link between the structure of a model and its continuous dynamics via the influence graph corresponding to the Jacobian matrix. Chapter 4 goes further in that direction by rendering explicit the formal relationships between the different semantics given to a model, from boolean to continuous or stochastic, and even the influence graph seen as a type. This goes to demonstrate the role of backbone that the structural model plays, with respect to the different views of the system, and partially explains how it is possible to deduce so much information (conservation laws, existence of multiple steady states, etc.) purely from a naked structural model.

The next section will provide some reminders on the different formalisms used throughout this manuscript.

1.2 Preliminaries - Reaction models as bipartite graphs

Starting from the inspiring use of the π-calculus in [54] to model signaling pathways in a cell, there has been a large amount of work around process calculi and of their stochastic extensions [29, 52] to formalize biochemical interactions. These stochastic versions allowed to link with the usual mathematical biology view of a system of ordinary differential equations (ODEs). However most of these formalisms only bring simulation tools to the modeler. On the other hand, the modeling of gene-regulatory networks through influence graphs along the work of Thomas [61] provides more global analyses, like the existence of cyclic attractors, but often impossible to use for large post-transcriptional regulation networks.

The Biochemical Abstract Machine [26] (`http//lifeware.inria.fr/BIOCHAM/`) was built as a simplification of the process algebras, using instead a rule-based language focused on reactions to describe biological systems. This point of view is shared with all the Systems Biology Markup Language (SBML) [42] community (see for instance databases like `reactome.org`, KEGG [44] or `biomodels.net`) and enables the exchange of models but also, as we shall see, the possibility to reason at different levels on a same model. BIOCHAM also adds the use of Temporal Logics as a second formalism

to encode the expected or observed properties of the system, from a purely qualitative view to a completely quantitative one. This allows us to automatically check that the model behaves as specified through model-checking tools adapted to the considered level. As stated above, in this manuscript we will however focus on the reaction/species part of the models and not on the specification, which leads to another commonly used formalism.

The use of Petri-nets to represent those reaction models, taking into account the difference between compounds and reactions in the graph, and making available various kinds of analyses is quite old [53], however it has remained somehow focused towards mostly qualitative properties. While the structural focus will be shared in our work, we shall demonstrate that it does not restrict the analyses to purely qualitative ones.

1.2.1 Boolean and bounded views

The simplest view one can have of a system of reactions is purely qualitative and relies on a boolean (presence/absence) semantics. For systems like Kohn's map of the mammalian cell cycle regulation [46], with about 500 compounds and 800 reactions but very little quantitative information, this is the natural level, hence the formalism of Molecular Interaction Maps, developed by Kohn himself twelve years later [47]. It is also the choice made in the Pathway Logic of [31].

The parallel with electronic circuits becoming obvious when looking at the drawing of the map, the same tools can be applied with certain success: the reactions define a concurrent transition system on which model-checking is able to verify very efficiently some quite complex properties. For instance, that the original map, as published, does not provide any synthesis reaction for cyclin B.

We also turned this method into a machine learning system where a model not verifying a specification can be automatically revised into one that does [1].

Moreover under simple hypotheses on the possible kinetics of each reaction (and verified by all usual kinetic laws, like Mass Action, Michaelis Menten or Hill kinetics for instance) [7] it is possible to automatically derive the influence graph between compounds from the reaction model. This result linking formally reaction graph and influence graph permits us to benefit from (and improve on, as will be shown in Section 2.4) the known necessary conditions for multi-stability or oscillations proven in that context, from a reaction network with very little knowledge on the kinetics, and especially no hypothesis of linearity. This approach is quite complementary with Feinberg's Chemical Reaction Network Theory (CRNT) [33], which also relies on the reaction graph, especially in its recent developments (e.g., [22, 23, 34, 59]) to deduce information about the continuous dynamics of a Mass Action system.

The same kind of view, but with an integer number of compounds, can be applied to smaller models, leading to the classical Petri-net representation of the system, places corresponding to compounds, with amounts as tokens, and transitions to reactions in an immediate way [53]. Once again model-checking can be used to ensure the reachability of some states. This level will be detailed further in Section 1.2.3 and invariant computation, described in Chapter 2 will use it as a means to extract quantitative as well as qualitative information from the structure of the model. There is also information lying

between quantitative and qualitative, like the definition of modules from T-invariants as explored recently in [36, 38].

1.2.2 Continuous and Stochastic views

Associating rates or kinetics to each reaction, one can view the system at a stochastic level, with simulation of the corresponding continuous time Markov chain thanks to Gillespie's algorithm(s) [37] and stochastic model-checking.

For efficiency reasons, when the number of compounds considered is big enough but not infinite (unbounded polymerization), the continuous view of that same system, with ODEs derived automatically from the reactions, is preferred. If the dimension is small enough, mathematical tools like bifurcation theory will bring results about ranges of parameters for which a specific dynamical behavior can be obtained.

In any case, we can use simulations as a basis for continuous model-checking [1, 21] to once again provide automatic verification that the model behaves as specified for either high dimensional systems or for properties outside of the usual scope (e.g., properties about the maximal concentration reached by a transitory peak of the system and its time frame).

Recent works generalizing this model-checking step to constraint-solving allowed us to define a continuous degree of satisfaction of a specification by a model. Using it as a fitness function one can apply state of the art optimization techniques to obtain parameter learning with respect to both qualitative and quantitative information coded as a specification [14]. The same technique also allows to define some new notion of robustness, with respect to a given temporal logic specification.

1.2.3 Petri nets

Since as we have shown many of these views can be represented naturally using Petri nets, many of the articles included in this manuscript do use such formalism. We will here recall the most basic notations we used.

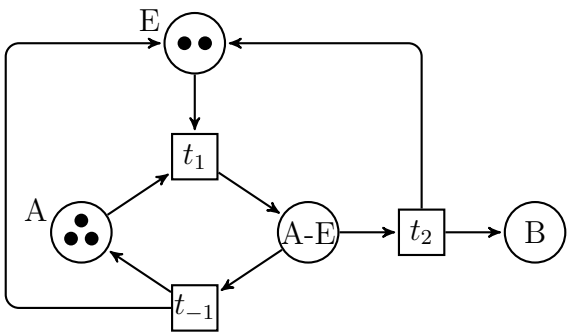


Figure 1.3: Biochemical model of Example 1, represented as a PN with a marking enabling t_1

A Petri net is a bipartite oriented multigraph of transitions, usually represented as

square boxes, and places, usually represented as circles, that defines a transition relation on *markings* of the net, i.e., multisets of tokens associated to places. The relation is defined by *firings* of transitions, i.e., when there are tokens (as many as the weights of the incoming arcs) in all pre-places of a transition, they can be consumed and as many tokens as the weights on the outgoing arcs are added to each post-place.

Example 1 *The classical enzymatic reaction written*[2] $A + E <=> A{-}E => B + E$ *corresponds to the Petri net depicted in Figure 1.3.*

In this Petri-net, starting from a marking with at least one token in A and in E, one can remove one of each to produce one token in A-E (firing of t_1*) and then either remove it to add again one token to A and one to E (firing of* t_{-1}*), or to add one B and one E (firing of* t_2*).*

Reaction models can usually be easily represented as PNs by mapping compounds/species to places and reactions to transitions (stoichiometry corresponding to the weights of the arc between the places and the transition), see Figure 1.

[2] in BIOCHAM-like syntax [2]

Prediction is very difficult, especially about the future.

Niels Bohr

Chapter 2

Dynamical Analysis Based on Structural Properties

Summary

2.1 Context

One of the most natural ways to try and tackle the lack of precise quantitative information for Systems Biology models, is to use the discrete structure that embodies the knowledge of the biologists to infer properties of the underspecified dynamical system. Indeed, even from the informal drawing that is still the most common means of explanation found in experimental papers, one can derive a formal object, a reaction model, for instance as a Petri net. These objects can then provide enough information in their multigraph structure to allow answering some questions about the possible dynamical behavior that would be obtained if kinetic parameters and rate laws were known. This information is also very general, whatever the intended dynamical semantics of the model since, as already noticed by Érdi and Tóth:

> The skeleton-like character of stoichiometry should be emphasised: it does not depend on the type of model (discrete or continuous, deterministic or stochastic […]) that is used afterwards when describing the time evolution. [32, Chapter 3, *Stoichiometry: the algebraic structure of complex chemical reactions*]

The two main schools that have emerged in this field, especially when applied to biochemical system, are probably the Stoichiometric Network Analysis (SNA) popularized by Schuster [40] and the Chemical Reaction Network Theory (CRNT) of Feinberg [33].

In this chapter, the first two articles can be related to SNA:

[16] Sylvain Soliman. "Invariants and Other Structural Properties of Biochemical Models as a Constraint Satisfaction Problem". In: *Algorithms for Molecular Biology* 7.15 (May 2012). DOI: 10.1186/1748-7188-7-15

[13] Faten Nabli and Sylvain Soliman. "Steady-state solution of biochemical systems, beyond S-Systems via T-invariants". In: *CMSB'10: Proceedings of the 8th International Conference on Computational Methods in Systems Biology*. Ed. by Paola Quaglia. CoSBi. Trento, Italy: ACM, Oct. 2010, pp. 14–22. ISBN: 978-1-4503-0068-1. DOI: 10.1145/1839764.1839768

The third one is focused on another structural property, more common in the Petri net community, but already applied to biochemical systems by Schuster himself [63] and more recently in [39], namely siphons and traps. It focuses on their computation by constraint programming and SAT solvers, and has been accepted in Constraints (it is actually an extended version of [11] published in *Principles and Practice of Constraint Programming 2012*, Springer-Verlag).

[12] Faten Nabli, Thierry Martinez, François Fages, and Sylvain Soliman. "On Enumerating Minimal Siphons in Petri nets using CLP and SAT solvers: Theoretical and Practical Complexity". In: *Constraints* 21.2 (2016), pp. 251–276. ISSN: 1383-7133. DOI: 10.1007/s10601-015-9190-1

Finally, we describe a paper closer to the CRNT approach. It improves the ten years old result by Christophe Soulé proving Thomas' hypotheses for multistationarity in the continuous setting [60].

[17] Sylvain Soliman. "A stronger necessary condition for the multistationarity of chemical reaction networks". In: *Bulletin of Mathematical Biology* 75.11 (Nov. 2013), pp. 2289–2303. DOI: 10.1007/s11538-013-9893-7

2.2 Place and Transition Invariants

The biggest successes obtained from SNA are probably the analysis of huge metabolic networks in terms of Elementary Modes, i.e. a basis for the cone of steady fluxes. There are however many other applications for these modes and the closely related Petri-net notion of transition invariants (T-invariants) or their dual, place invariants (P-invariants).

For instance, P-invariants provide a way to compute conserved moieties of a system, whatever the dynamical semantics and the precise kinetic coefficients and rate laws. Such property can be used of course to reduce the system, but also in other ways as will be shown in Section 3.3.

2.2.1 Computing invariants with Constraint Programming

In [15] and later [16] —included hereafter— we explored a way to compute such basis using Constraint Programming. These works are contemporary and complementary to the recent approaches using Mixed Integer Programming (MIP) to solve the same problem, especially when one wants to add side constraints like enumerating only small Elementary Modes [35]. We are still further investigating this line of research, especially in the context of finding modes that include some given transitions, which makes the usual conditions of minimality break. Note that this method is quite orthogonal to the use of constraints to compute the Generating Flux Modes, optionally containing some reactions of interest, adding thermodynamical restrictions or enforcing optimal yield, as done in [48, 49, 55]. Globally theses techniques show the current interest in using symbolic and high-level approaches to tackle the computational complexity of this hot topic.

[16] Sylvain Soliman. "Invariants and Other Structural Properties of Biochemical Models as a Constraint Satisfaction Problem". In: *Algorithms for Molecular Biology* 7.15 (May 2012). DOI: 10.1186/1748-7188-7-15

Soliman *Algorithms for Molecular Biology* 2012, **7**:15
http://www.almob.org/content/7/1/15

AMB ALGORITHMS FOR MOLECULAR BIOLOGY

RESEARCH Open Access

Invariants and Other Structural Properties of Biochemical Models as a Constraint Satisfaction Problem

Sylvain Soliman

Abstract

Background: We present a way to compute the minimal semi-positive invariants of a Petri net representing a biological reaction system, as resolution of a Constraint Satisfaction Problem. The use of Petri nets to manipulate Systems Biology models and make available a variety of tools is quite old, and recently analyses based on invariant computation for biological models have become more and more frequent, for instance in the context of module decomposition.

Results: In our case, this analysis brings both qualitative and quantitative information on the models, in the form of conservation laws, consistency checking, etc. thanks to finite domain constraint programming. It is noticeable that some of the most recent optimizations of standard invariant computation techniques in Petri nets correspond to well-known techniques in constraint solving, like symmetry-breaking. Moreover, we show that the simple and natural encoding proposed is not only efficient but also flexible enough to encompass sub/sur-invariants, siphons/traps, etc., i.e., other Petri net structural properties that lead to supplementary insight on the dynamics of the biochemical system under study.

Conclusions: A simple implementation based on GNU-Prolog's finite domain solver, and including symmetry detection and breaking, was incorporated into the BIOCHAM modelling environment and in the independent tool Nicotine. Some illustrative examples and benchmarks are provided.

1 Background

1.1 Introduction

Reaction models like those of `reactome.org`, KEGG pathway database [1] or `biomodels.net` represent a growing part of Systems Biology especially for metabolic or signalling pathways, cell-cycle and more generally post-genomic regulation systems. They build on established standards like BioPAX or SBML [2] to facilitate the exchange and comparison of models and benefit from a large number of available tools, especially ODE integration based simulators.

The use of Petri nets to represent those models, taking into account the difference between compounds and reactions in the graph, and make available various kinds of analyses is quite old [3], however it remains somehow focused towards mostly qualitative and structural properties. Some have been used for module decomposition, like (I/O) T-invariants [4,5], related to dynamical notions of elementary flux modes [6]. However, there is, to our knowledge, very little use of P-invariant computation, which provides both qualitative information about some notion of module related to the "life cycle" of compounds, and quantitative information related to conservation laws - each P-invariant defines a conserved moiety of the obtained ODE system, whatever the rate laws - and Jacobian matrix singularity - induced by any P-invariant since it defines a linear dependency between variables. Conservation law extraction is actually already provided by a few tools, but then using numerical methods, based on the quantitative view of the model, and not integer arithmetic (as in direct P-invariant analysis).

We present here a very simple way to incorporate invariant computation in an existing biological modelling tool, using constraint programming with symmetry detection and breaking. We compare it to other approaches and

Correspondence: Sylvain.Soliman@inria.fr
Equipe-Projet Contraintes, INRIA Paris-Rocquencourt, BP105, 78153 Le Chesnay Cedex, France

evaluate it, for the case of P-invariants, on some examples of various sizes, like the MAPK cascade models of [7] and [8]. This experimentation is done through an implementation of the described method in the BIOCHAM modelling environment [9,10], and in the independent tool Nicotine. We benchmark the efficiency against state of the art Petri net tools on various models. Finally we show that the presented approach allows to compute, within the same framework, other interesting structural properties like sub/sur-invariants or siphons/traps, bringing even more insight into the dynamics of the biochemical system under study.

1.2 Petri net view of a reaction model

A Petri net is a bipartite oriented (weighted) graph of transitions, usually represented as square boxes, and places, usually represented as circles, that defines a (actually not unique) transition relation on *markings* of the net, i.e., multisets of tokens associated to places. The relation is defined by *firings* of transitions, i.e., when there are tokens (as many as the weight of the incoming arc) in all pre-places of a transition, they can be consumed and as many tokens as the weight on the outgoing arc are added to each post-place. The classical Petri net view of a reaction model is simply to associate biochemical *species to places* and biochemical *reactions to transitions*.

Example 1 *For instance the enzymatic reaction written (in BIOCHAM-like syntax),* `A + E ⇔ A-E ⇒ B + E` *corresponds to the following Petri net* (Figure1)

In this Petri net, starting from a marking with at least one token in A and in E, one can remove one of each to produce one token in A-E (firing of t_1) and then either remove it to add again one token to A and one to E (firing of t_{-1}), or to add one B and one E (firing of t_2).

P (resp. T) invariants are defined, as usual, as vectors V representing a multiset of places (resp. of transitions) such that $V \cdot I = 0$ (resp. $I \cdot V = 0$) where I is the incidence matrix of the Petri net, i.e. I_{ij} is the number of arcs from transition i to place j, minus the number of arcs from place j to transition i. Intuitively, a P-invariant is a multiset representing a weighting of the places and such that any such weighted marking remains invariant by any firing; a T-invariant represents a multiset of firings that will leave invariant any marking (see also section 2.1). As explained in the introduction, for reaction models these invariants are used for flux analysis, variable simplification through conservation law extraction, module decomposition, etc. Note that we are concerned with the classical invariant problem and thus restrict our study to integer weights. This is an important difference with respect to the aforementioned flux analyses but it arises from the fact that the biochemical models we studied did not come from metabolism but from the modelling of signal transduction pathways, cell cycle, circadian rhythm, etc. In all these cases the stoichiometry was integer and, for instance, the extracted conservation laws will include only integer number of molecules.

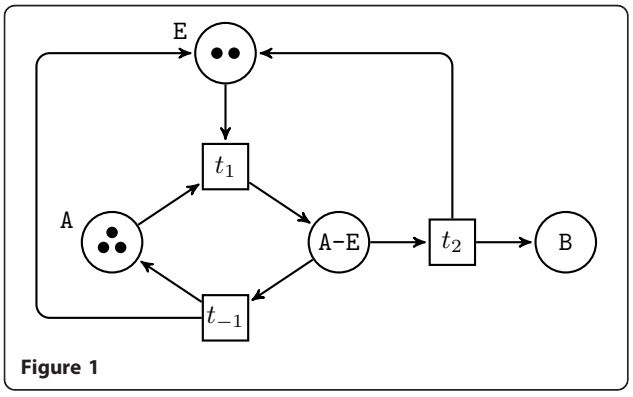


Figure 1

1.3 Related work

To compute the invariants of a Petri net, especially if this computation is combined with other Petri net analyses, like sinks and sources, traps, deadlocks, etc. the most natural solution is to use a Petri net dedicated tool like INA, PiNA, or Charlie for instance through the interface of Snoopy [11], which allows the import of SBML models as Petri nets. Standard integer methods like Fourier-Motzkin elimination will then provide an efficient means to compute P or T-invariants (see for instance [12] for a review). These methods however generate lots of candidates which are afterwards eliminated and also need to incorporate some means (like equality class definition) to avoid combinatorial explosion at least in some simple cases, as explained in Section 2.2.

Another way to extract the minimal semi-positive invariants of a model is to use one of the software tools that provide this computation for biological systems, generally as "conservation law" computation, and based on linear algebra methods like QR factorization [13]. This is the case for instance of the METATOOL [14] and COPASI [15] tools. The idea is to use a linear relaxation of the problem, which suits well very big graphs, but needs again *a posteriori* filtering of the candidate solutions. Moreover, these methods do not incorporate any means of symmetry elimination (see Section 2.2). A recent technique for elementary mode computation relies on Mixed Integer Programming (MIP) [16] and is thus quite similar in theory to the ideas of thus article, however it is tailor-made for elementary modes whereas for invariants pure Integer Programming would be enough, it is focused around the computation of a partial basis of these modes, which is an important problem but not the focus in this article, and - once again - it does not incorporate any symmetry breaking.

Finally, the most recent developments in invariant computation rely on a symbolic encoding through Binary decision Diagrams [17]. The tools based on this technique can prove quite efficient and are not unrelated to the symbolic encoding we present here through constraints. However they do not seem to integrate symmetry detection, also rely on filtering for minimality and thus, though they provide a symbolic solution very fast in some cases, might also benefit from some of the ideas we present. See section 2.5 for a more precise evaluation.

2 Results and Discussion

2.1 Finding invariants as a Constraint Satisfaction Problem

We will illustrate our new method for computing the invariants with the case of P-invariants (but T-invariants, being dual, would work in the same fashion). Consider a Petri net with p places and t transitions, these transitions represent reactions $L_i \rightarrow R_i$, where L_i encodes the stoichiometry of the reactants as a vector over places, and R_i the same for the products of the reaction. A P-invariant is a vector $V \in \mathbb{N}^p$ s.t. $V^T \cdot I = 0$, i.e. $\forall 1 \leq i \leq t\ V \cdot L_i = V \cdot R_i$. Since those vectors all live in $\mathbb{N}^p$, it is quite natural to see this as a Constraint Satisfaction Problem (CSP) [18-20] with t (linear) equality constraints on p finite domain (FD) variables.

Example 2 *Using the Petri net of Example 1 we have:*

$$\mathtt{A} + \mathtt{E} \Rightarrow \mathtt{A} - \mathtt{E}$$
$$\mathtt{A} - \mathtt{E} \Rightarrow \mathtt{A} + \mathtt{E}$$
$$\mathtt{A} - \mathtt{E} \Rightarrow \mathtt{B} + \mathtt{E}$$

This results in the following equations:

$$A + E = AE \quad (1)$$

$$AE = A + E \quad (2)$$

$$AE = B + E \quad (3)$$

where obviously equation (2) is redundant.

The task is actually to find invariants with minimal support, with respect to set inclusion (a linear combination of invariants belonging to $\mathbb{N}^p$ also being an invariant), i.e., having as few non-zero components as possible, these components being as small as possible, but of course non trivial, we thus add the constraint that $V \cdot \mathbf{1} > 0$.

Example 3 *In our running example we thus add* $A + E + AE + B > 0$.

Now, to ensure minimality the labelling is invoked from small to big values. This means that for each variable, if an enumeration remains necessary after constraint propagation, values are tried in an increasing order starting at 0. This is closely related to the enumeration strategy used in the mixed integer programming method of [16] that allows them to look for *shortest* elementary modes. Such a restriction in the construction of the basis might thus also be possible in our approach. Then, a branch and bound procedure is wrapped around this search for solutions, maintaining a partial base $\mathcal{B}$ of P-invariant vectors and adding the constraint that a new vector V is solution if $\forall B \in \mathcal{B} \prod_{B_i \neq 0} V_i = 0$, which means that its support is not bigger than that of any vector of the base.

Unfortunately, even with the last constraint, no search heuristic was found that makes removing subsumed P-invariants unnecessary. Thus, if a new vector is added to $\mathcal{B}$, previously found vectors with a bigger support must be removed. Section 2.6 will demonstrate other structural properties for which this step is not necessary.

The algorithm can be summarized as follows:

Algorithm 1 Minimal invariants computation

1: post the CSP for invariant V: $\forall 1 \leq i \leq t\ V \cdot L_i = V \cdot R_i$ and $V \cdot \mathbf{1} > 0$
2: **repeat**
3: find a solution, enumerating from low to high
4: add the solution to the basis
5: remove non-minimal invariants from the basis if there are any
6: post the new constraint $\forall B \in \mathcal{B} \prod_{B_i \neq 0} V_i = 0$
7: **until** no solution found
8: expand symmetrical solutions of $\mathcal{B}$

This algorithm was implemented directly into Nicotine[1] and then added to BIOCHAM [9], which are both programmed in GNU-Prolog, and allowed for immediate testing.

Example 4 *In our running example we find two minimal semi-positive P-invariants:*

- $E = AE = 1$ *and* $A = B = 0$
- $A = B = AE = 1$ *and* $E = 0$

2.2 Equality classes

The problem of finding minimal semi-positive invariants is clearly EXPSPACE-hard since there can be an exponential number of such invariants. For instance the model given in Example 5 (described in [12] among others, and called "classic X-Y" in [17], where × is the number of places between each pair of transitions and Y the number of transitions) has 2^n minimal semi-positive P-invariants (each one with either A_i or B_i equal to 1 and the other equal to 0).

Example 5 (Classic 2-n) (Figure2)

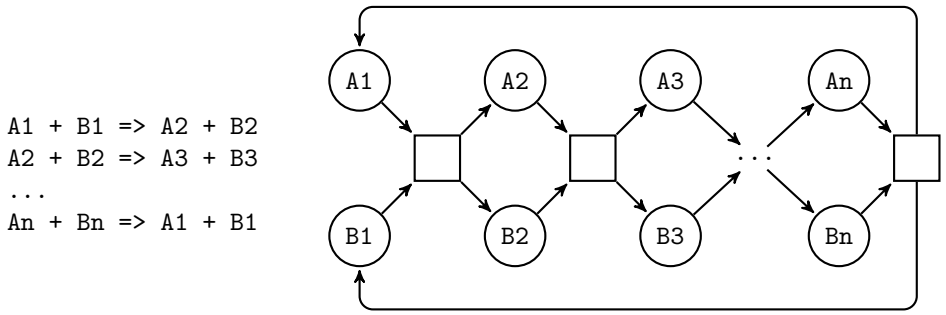


Figure 2

A first remark is that in this example, there is a variable symmetry between all the pairs (A_i, B_i) of variables corresponding to places. This symmetry is easy to detect (purely syntactical) and can be eliminated through the usual ordering of variables, by adding the constraints $A_i \leq B_i$.

This classical CSP optimization is enough to avoid most of the trivial exponential blow-ups and corresponds to the initial phase of *parallel places* detection and merging of the equality classes optimization [21] for the standard Fourier-Motzkin algorithm. Note however that in that method, classes of equivalent variables are detected and eliminated before and *during* the invariant computation, which would correspond to local symmetry detection and was not implemented in our prototype.

Moreover, in [21], *equality class* elimination is done through replacement of the symmetric places by a representative place. The full method reportedly improves by a factor two the computation speed. Even if in the context of the original article this is done only for ordinary Petri nets (Petri nets where the weights are only 0 or 1), we can see that it can be even more efficient to use this replacement technique in our case:

Example 6

...

A + B ⇒ 4*C

...

Instead of simply adding $A \leq B$ to our constraints, which will lead to 3 solutions when $C = 1$ before symmetry expansion: $(A, B) \in \{(0, 4), (1, 3), (2, 2)\}$, *replacing A and B by D will reduce to a single solution $D = 4$ before expansion of the subproblem $A + B = D$.*

This partial detection of independent subproblems, which can be seen as a complex form of symmetry identification, can once again be done syntactically at the initial phase, and can be stated as follows: replace $\sum_i k_i * A_i$ by a single variable A if all the A_i occur only in the context of this sum i.e., in our Petri net all pre-transitions of A_i are connected to A_i with k_i edges and to all other A_j with k_j edges and same for post-transitions. For a better constraint propagation, another intermediate variable can be introduced such that $A = gcd(k_i) \cdot A'$. In our experiments the simple case of *parallel places* (i.e., all k_i equal to 1 in the sum) was however the one encountered most often.

2.3 Example, the MAPK Cascade

The MAPK signal transduction cascade is a well studied system that appears in lots of organisms and is very important for regulating cell division [22]. It is composed of layers, each one activating the next, and in detailed models shows two intertwined pathways conveying EGF and NGF signals to the nucleus.

A simple MAPK cascade model, that of [23] without scaffold, is used here as an example to show the results of P-invariant computation.

Seven minimal semi-positive P-invariants are found almost instantly. Intuitively, they represent the different levels of the cascade (i.e., RAFK, RAF, MEK and MAPK) and the corresponding phosphatases (RAFPH, MEKPH and MAPKPH). The use of those P-invariants as visual *modules*, as depicted in Figure 3 is quite similar to one part of the approach of [24] to make biochemical systems more easy to grasp. The full list is given in Table 1.

In the next section other examples are used as benchmarks of this method, they are all much bigger than this one, which had only about 30 compounds, however note that one of those is still a model of the MAPK signalling cascade.

Note that these 7 P-invariants define 7 algebraic conservation rules (i.e., mass conservation) and thus decrease the size of the corresponding ODE model from 22 variables and equations to only 15.

2.4 Evaluation on other biochemical examples

Schoeberl's model is a more detailed version of the MAPK cascade, which is quite comprehensive [8], but too big to be studied by hand. It can however be easily broken down into fourteen more easily understandable units formed by P-invariants, as shown in Table 2, along other examples representing amongst the biggest reaction networks publicly available.

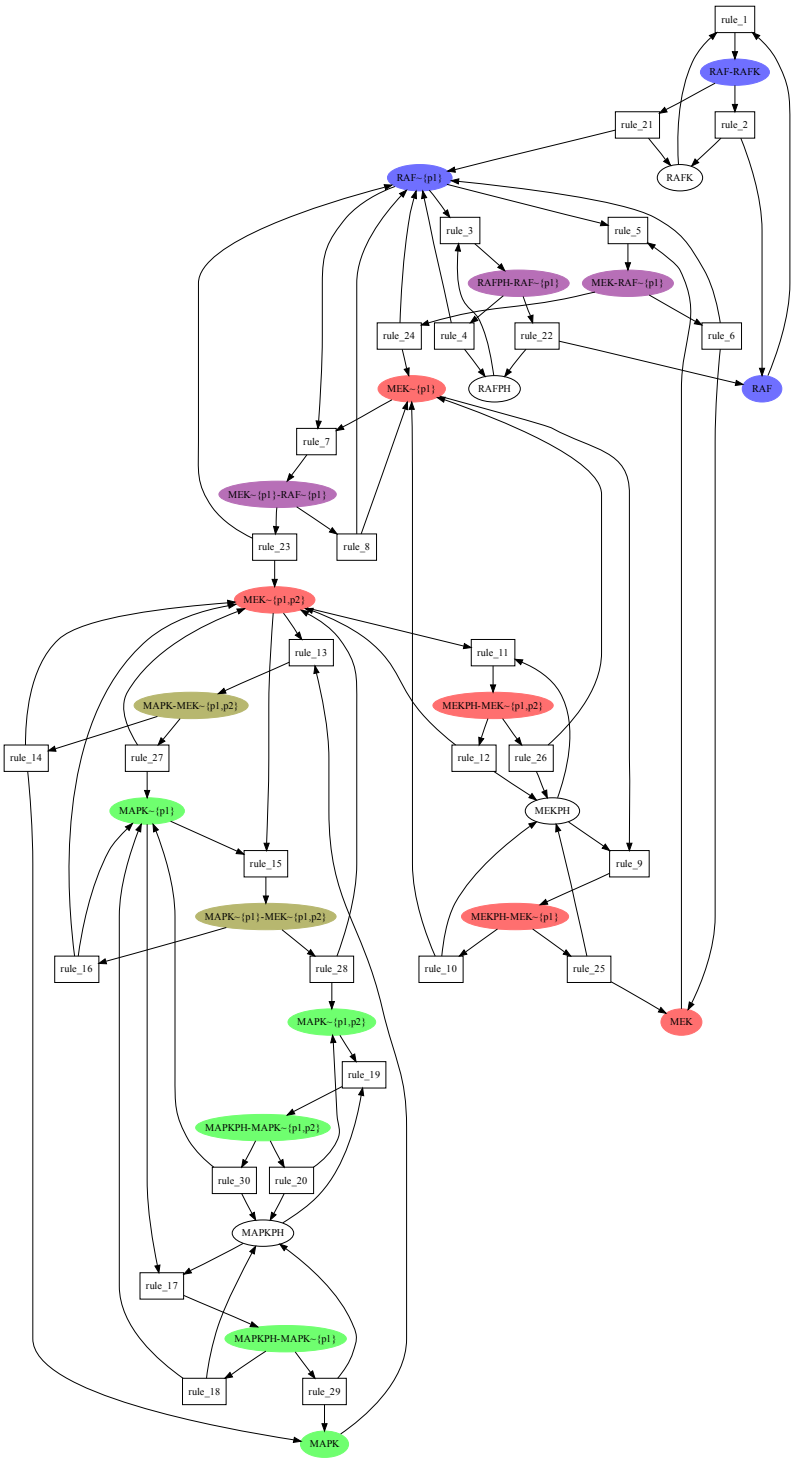


Figure 3 Some conservation laws of the MAPK model of [23]. 3 of the 7 P-invariants found in the MAPK cascade model of [23]. The blue one (RAF), the pink one (MEK) and the green one (MAPK) with intersections in purple (blue+pink) and khaki (pink+green).

All the curated models in the September 2010 release of `biomodels.net` were also tested and none of them required more than 1s to compute all its minimal P-invariants.

We could not compare our results with those provided in [13] since the models they use, coming from metabolic pathways flux analyses, do not have an integer stoichiometry matrix, however the examples of Table 2

Table 1 P-invariants of the MAPK cascade model of [23]

RAFK, RAF-RAFK
RAFPH, RAFPH-RAF~{p1}
RAF, MEK-RAF~{p1}, RAF-RAFK, RAFPH-RAF~{p1}, MEK~{p1}-RAF~{p1}, RAF~{p1}
MEKPH, MEKPH-MEK~{p1}, MEKPH-MEK~{p1, p2}
MEK, MAPK-MEK~{p1, p2}, MEK-RAF~{p1}, MEKPH-MEK~{p1}, MEKPH-MEK~{p1, p2}, MAPK~{p1}-MEK~{p1, p2}, MEK~{p1}-RAF~{p1}, MEK~{p1}, MEK~{p1, p2}
MAPKPH, MAPKPH-MAPK~{p1}, MAPKPH-MAPK~{p1, p2}
MAPK, MAPK-MEK~{p1, p2}, MAPKPH-MAPK~{p1}, MAPK~{p1, p2} MAPK~{p1}-MEK~{p1, p2}, MAPK~{p1}, MAPKPH-MAPK ~{p1, p2},

Full list of the P-invariants of the MAPK cascade model of [23]

show the feasibility of P-invariant computation by constraint programming for quite big networks. Note that for networks of this size, the upper bound of the domain of variables had to be set manually. It was actually set to the value 8, which is about the double of the maximum value in all the biological models we have encountered up to now. The only over-approximation of the upper bound found was the product of the *l.c.m.* of stoichiometric coefficients of each reaction, which explodes really fast and leads to unnecessarily long computation. The manual bound results in a loss of completeness, but it is not enforced either by QR-factorization methods, and does not seem to miss anything on real life examples.

Though they are not specifically suited for this task (i.e., finding integer invariants), we tried some of the most well known Elementary Flux Modes computing packages on these examples. METATOOL [14] and efmtool [25] were chosen, since both can be run as Matlab packages. The results are not included in Table 2 but are summarized with the non-biochemical examples of next section.

2.5 Non-biochemical benchmarks

Even if our main purpose is to use the insight on the dynamics gained from the structural properties computed by our CSP, an evaluation of the proposed method on non-biochemical models remains of interest.

The literature on invariant computation is quite large, however there does not seem to exist any standardized benchmark. Each author selects some examples with different properties (see for instance [12] from which only a few examples are used in [17], even though it is cited as reference) and few reuse the previously published sets of examples.

Moreover, even when the software used in these articles is available, usually only binary implementations are available, and only for some specific architectures and through a specific request process. In some cases none is provided at all.

Therefore, using a machine comparable in specifications, we chose to reuse the data published in the most recent work, that of Ciardo et al. [17]. Since we had to re-encode ourselves the selected examples, only a subset of their benchmarks is covered, namely the classical dining philosophers problem [26], the standard exponential invariant case [12] and the circular trains [27]. These seem to cover the whole range of different schemes appearing in [17].

Note that there are usually many symmetries in these parametric examples and thus that a more powerful (or manual) symmetry detection would be called for in these specific cases. Nevertheless, since (intracellular) biochemical systems usually do not generate such structure, we did not push further the integration of more advanced symmetry detection/breaking in our tools.

All the models used for the biochemical and non-biochemical benchmarks can be found at: http://contraintes.inria.fr/~soliman/nicotine_data/

METATOOL's "CONSERVATION RELATIONS" were used when possible, but that only allows to find - as expected - 91 out of the 10 billion invariants for the classic example, in 0.33s. Models were thus *transposed* such that METATOOL and efmtool's EFM search correspond to P-invariant computation. Transposed models appear with a 'b' ending in the data repository. efmtool was given the SBML files as input whereas some `.dat` files were generated for METATOOL. For all the examples of this section as well as Kohn's map, METATOOL

Table 2 Minimal semi-positive P-invariant computation on bigger models of biochemical reaction networks

Model	transit.	places	P-invar.	time (s)	Invariant size
Schoeberl's MAPK [8]	125	105	13	0.53	from 2 to 44
Calzone et al. E2F/Rb [31]	~500	~400	79	18	from size 1 (EP300) to about 230 (E2F1 box)
Kohn's map [32]	~800	~500	70	171	from size 1 (Myt1) to about 200 (pRb or cdk2)

Minimal semi-positive P-invariant computation on bigger models of biochemical reaction networks

gave the error message "Cannot sort modes with more than 52 rows" that was interpreted as some kind of "out of memory" error. For efmtool, in the same cases (all examples of this section plus Kohn's map) the computation was stopped after 10 minutes or more, with messages like "iteration 43/116: 224850 modes, dt = 2040206ms." that were interpreted as overtime. Note however that as already stated, these packages do not focus on *integer* stoichiometric matrices and thus have a much broader scope that might explain their poor performance on our benchmarks.

The results are presented in Table 3, where as in [17] "om" represents an out-of-memory error, and "ot" an overtime. "na" was used when conservation relations are strictly fewer than P-invariants. The results seem to indicate that a constraint-based approach fares reasonably well, usually in the same order of magnitude as some purely symbolic encoding via decision diagrams [17], whereas the solutions of the CSP are explicit. Even in the case where finding explicit solutions revealed too costly (classic 10-10, which has 10^{10} minimal P-invariants), one can stop the computation before symmetry expansion and get an answer in a reasonable time.

The CSP approach can therefore be seen as a kind of intermediate between purely implicit (i.e., solutions encoded, for instance as a decision diagram, and needing to be decoded to be displayed) and purely explicit methods. It also remains very flexible as next section will prove and could incorporate many more optimizations (variable ordering heuristics, more symmetry elimination, etc.) at a quite low cost.

All the 80 Petri nets of http://www.petriweb.org/ were also tested. Only one took more than 1s: model 1516, which took about 3s to compute 1133 minimal P-invariants. Since we do not have data for the other approaches on these models they were not added to the table of results but they confirm the feasibility and generality of our approach.

We think that the structure of this kind of net is however very different (average degree, arc weights, etc.) from that of usual biochemical reaction models and intend to explore this distinction further in the future.

2.6 Generalizing the approach to other structural properties

An interesting feature of the presented method is that it is actually flexible enough to encompass other structural properties than place or transition invariants. This is, to our knowledge, not the case of other alternative methods.

If for the Petri net of Example 1 one obtained the constraints shown in Example 2 to compute P-invariants, one can notice that they can easily be adapted to compute sur- or sub-invariants, i.e., weighted sums that can only grow (resp. decrease) during the evolution of the system (see [28], for instance, for a formal definition). Indeed the following CSP describes exactly all the sub-invariants of the system and is obtained in the same manner but with ≤ instead of =.

Example 7 *Using the Petri net of Example 1:*

$$\mathtt{A} + \mathtt{E} \Rightarrow \mathtt{A} - \mathtt{E}$$

$$\mathtt{A} - \mathtt{E} \Rightarrow \mathtt{A} + \mathtt{E}$$

$$\mathtt{A} - \mathtt{E} \Rightarrow \mathtt{B} + \mathtt{E}$$

results in the following FD constraints:

$$A + E \leq AE \tag{4}$$

$$AE \leq A + E \tag{5}$$

$$AE \leq B + E \tag{6}$$

Sur-invariants would be obtained with ≥ instead of ≤.

Now, getting a basis of minimal sub/sur-invariants can be done with the same branch and bound technique used for invariants, allowing to obtain information on pools of species of the biochemical system that, for instance, never increase during any ODE simulation.

One can go slightly farther and once again reuse the same machinery, including symmetry breaking, to compute siphons and traps of the Petri net (see [29] for definition and example of use in biology). This time a boolean CSP is obtained with the following constraints for the example of traps:

Example 8 *Using the Petri net of Example 1 we obtain the following boolean constraints:*

$$A \vee E \Rightarrow AE \tag{7}$$

Table 3 Minimal semi-positive P-invariant computation on general (non-biochemical) benchmarks of the literature

model	BDD V2	BDD V4	GreatSPN	Nicotine	Metatool CR	Metatool EFM	efmtool
trains 10-10	4.81	om	0.03	3.26	na (20)	om	ot
classic 10-10	0.01	0.01	ot	0.15	na (91)	om	ot
philo 30	1.04	0.01	0.01	2.68	3.04	om	ot

Minimal semi-positive P-invariant computation on general (non-biochemical) benchmarks of the literature. Times are given in seconds. BDD V2 and V4 (implicit) and GreatSPN (explicit) performances as per [17]. Note that for the *classic* example, time was measured for Nicotine before symmetry expansion (semi-implicit) since there are 10^{10} explicit solutions.

$$AE \Rightarrow A \vee E \tag{8}$$

$$AE \leq B \vee E \tag{9}$$

To compute siphons one simply need to reverse $\Rightarrow$ into $\Leftarrow$.

Note that in the boolean domain, the support minimality can be imposed by enumerating in increasing (lexicographic) order, there is no need for any a *posteriori* check of minimality (step 5 of Algorithm 1). The algorithm thus becomes:

Algorithm 2 Minimal traps computation
1: post the CSP for trap V
2: **repeat**
3: find a solution, enumerating from low to high
4: add the solution to the basis
5: post the new constraint $\forall B \in \mathcal{B} \prod_{B_i \neq 0} V_i = 0$
6: **until** no solution found
7: expand symmetrical solutions of $\mathcal{B}$

This computation of traps and siphons can actually bring information about the dynamics of the model, including temporal logic formulae that it satisfies[2], together with other structural properties [4,30] they provide an interesting toolkit to analyze structurally the dynamics of a Systems Biology model.

3 Conclusion

P-invariants of a biological reaction model are not so difficult to compute in most cases. They carry information about conservation laws that are useful for efficient and precise dynamical simulation of the system, and provide some notion of module, which is related to the life cycle of molecules. T-invariants are already used more commonly, and get more and more focus recently.

We introduced a new method to efficiently compute P and T-invariants of a reaction network, based on FD constraint programming. It includes symmetry detection and breaking and scales up well to the biggest reaction networks found. Completeness is lost on the biggest examples but we still look for a better upper bound on domains to restore it.

The idea of applying constraint based methods to classical problems of the Petri net community is not new, but seems currently mostly applied to the model-checking. We argue that structural problems (invariants, sinks, attractors, etc.) can also benefit from the know-how developed for finite domain CP solving, like symmetry breaking, search heuristics, flexibility, etc. and thus intend to generalize our approach to other problems of this category.

Endnotes

[1]http://contraintes.inria.fr/~soliman/nicotine.html

[2]This is the topic of a paper currently submitted to the CMSB 2011 conference. Depending on the outcome, a reference or a short explanation will be added.

Acknowledgements
We thank the French ANR project BioTempo (ANR-10-BLANC-0218) for its support.

Competing interests
The author declares that they have no competing interests.

Received: 29 April 2011 Accepted: 29 May 2012 Published: 29 May 2012

References

1. Kanehisa M, Goto S: **KEGG: Kyoto Encyclopedia of Genes and Genomes.** *Nucleic Acids Research* 2000, **28**:27-30.
2. Hucka M, *et al*: **The Systems Biology Markup Language (SBML): A Medium for Representation and Exchange of Biochemical Network Models.** *Bioinformatics* 2003, **19(4)**:524-531.
3. Reddy VN, Mavrovouniotis ML, Liebman MN: **Petri net representations in metabolic pathways.** In *Proceedings of the 1st International Conference on Intelligent Systems for Molecular Biology (ISMB)*. Edited by: Hunter L, Searls DB, Shavlik JW. AAAI Press; 1993:328-336.
4. Gilbert D, Heiner M, Lehrack S: **A Unifying Framework for Modelling and Analysing Biochemical Pathways Using Petri Nets.** In *CMSB'07: Proceedings of the fifth international conference on Computational Methods in Systems Biology, of Lecture Notes in Computer Science. Volume 4695*. Edited by: Calder M, Gilmore S. Edinburgh, Scotland: Springer-Verlag; 2007.
5. Grafahrend-Belau E, Schreiber F, Heiner M, Sackmann A, Junker BH, Grunwald S, Speer A, Winder K, Koch I: **Modularization of biochemical networks based on a classification of Petri net by T-invariants.** *BMC Bioinformatics* 2008, **9(90)**.
6. Schuster S, Fell DA, Dandekar T: **A general definition of metabolic pathways useful for systematic organization and analysis of complex metabolic networks.** *Nature Biotechnology* 2002, **18**:326-332.
7. Chickarmane V, Kholodenkob BN, Sauro HM: **Oscillatory dynamics arising from competitive inhibition and multisite phosphorylation.** *Journal of Theoretical Biology* 2007, **244**:68-76.
8. Schoeberl B, Eichler-Jonsson C, Gilles E, Muller G: **Computational modeling of the dynamics of the MAP kinase cascade activated by surface and internalized EGF receptors.** *Nature Biotechnology* 2002, **20(4)**:370-375.
9. Calzone L, Fages F, Soliman S: **BIOCHAM: An Environment for Modeling Biological Systems and Formalizing Experimental Knowledge.** *Bioinformatics* 2006, **22(14)**:1805-1807.
10. Fages F, Soliman S, Chabrier-Rivier N: **Modelling and Querying Interaction Networks in the Biochemical Abstract Machine BIOCHAM.** *Journal of Biological Physics and Chemistry* 2004, **4(2)**:64-73.
11. Rohr C, Marwan W, Heiner M: **Snoopy - a unifying Petri net framework to investigate biomolecular networks.** *Bioinformatics* 2010, **26(7)**:974-975.
12. Colom JM, Silva M: **Convex geometry and semiflows in P/T nets. A comparative study of algorithms for computation of minimal P-semiflows.** In *Advances in Petri Nets 1990, of Lecture Notes in Computer Science. Volume 483*. Edited by: Rozenberg G. London, UK: Springer-Verlag; 1991:79-112.
13. Vallabhajosyulaa RR, Chickarmane V, Sauro HM: **Conservation Analysis of Large Biochemical Networks.** *Bioinformatics* 2005, **22**:346-353.
14. von Kamp A, Schuster S: **Metatool 5.0: fast and flexible elementary modes analysis.** *Bioinformatics* 2006, **22(15)**:1930-1931.
15. Hoops S, Sahle S, Gauges R, Lee C, Pahle J, Simus N, Singhal M, Xu L, Mendes P, Kummer U: **COPASI - a COmplex PAthway SImulator.** *Bioinformatics* 2006, **22(24)**:3067-3074.
16. de Figueiredo LF, Podhorski A, Rubio A, Kaleta C, Beasley JE, Schuster S, Planes FJ: **Computing the shortest elementary flux modes in genome-scale metabolic networks.** *Bioinformatics* 2009, **25(23)**:3158-3165.
17. Ciardo G, Mecham G, Paviot-Adet E, Wan M: **P-Semiflow Computation with Decision Diagrams.** *PETRI NETS '09: Proceedings of the 30th International Conference on Applications and Theory of Petri Nets* Berlin, Heidelberg: Springer-Verlag; 2009, 143-162.

18. Mackworth AK: **Consistency in Networks of Relations.** *Artfcial Intelligence* 1977, **8**:99-118.
19. Meseguer P: **Constraint satisfaction problems: an overview.** *AI Communications* 1989, **2**:3-17.
20. Kumar V: **Algorithms for Constraint-Satisfaction Problems: A Survey.** *AI Magazine* 1992, **13**:32-44.
21. Law CF, Gwee BH, Chang J: **Fast and memory-efficient invariant computation of ordinary Petri nets.** *IEE Proceedings: Computers and Digital Techniques* 2007, **1(5)**:612-624.
22. Roovers K, Assoian RK: **Integrating the MAP kinase signal into the G1 phase cell cycle machinery.** *BioEssays* 2000, **22(9)**:818-826.
23. Levchenko A, Bruck J, Sternberg PW: **Scaffold proteins may biphasically affect the levels of mitogen-activated protein kinase signaling and reduce its threshold properties.** *PNAS* 2000, **97(11)**:5818-5823.
24. Hardy S, Robillard PN: **Visualization of the simulation data of biochemical network models: a painted Petri net approach.** In *Proceedings of the SCSC'07 summer computer simulation conference.* Edited by: San Diego, CA. USA: Society for Computer Simulation International; 2007:802-808.
25. Terzer M, Stelling J: **Large-scale computation of elementary flux modes with bit pattern trees.** *Bioinformatics* 2008, **24(19)**:2229-2235.
26. Graf S, Steffen B, Lüttgen G: **Compositional Minimisation of Finite State Systems Using Interface Specifications.** *Journal of Formal Aspects of Computing* 1996, **8(5)**:607-616.
27. Genrich H: **Predicate/transition nets.** In *Advances in Petri nets, of Lecture Notes in Computer Science. Volume 254.* Edited by: Brauer W, Reisig W, Rozenberg G. Springer-Verlag; 1987:207-247.
28. Sifakis J: **Structural Properties of Petri Nets.** In *Proceedings of MFCS'78, 7th Symposium on the Mathematical Foundations of Computer Science, of Lecture Notes in Computer Science. Volume 64.* Edited by: Winkowski J. Springer-Verlag; 1978:474-483.
29. Zevedei-Oancea I, Schuster S: **Topological analysis of metabolic networks based on Petri net theory.** *In Silico Biology* 2003, **3(29)**.
30. Nabli F, Soliman S: **Steady-state solution of biochemical systems, beyond S-Systems via T-invariants.** In *CMSB'10: Proceedings of the 8th International Conference on Computational Methods in Systems Biology.* Edited by: Quaglia P. CoSBi, ACM; 2010:14-22.
31. Calzone L, Gelay A, Zinovyev A, Radvanyi F, Barillot E: **A Comprehensive Modular Map of Molecular Interactions in RB/E2F Pathway.** *Molecular Systems Biology* 2008, **4(173)**.
32. Kohn KW: **Molecular Interaction Map of the Mammalian Cell Cycle Control and DNA Repair Systems.** *Molecular Biology of the Cell* 1999, **10(8)**:2703-2734.

doi:10.1186/1748-7188-7-15
Cite this article as: Soliman: Invariants and Other Structural Properties of Biochemical Models as a Constraint Satisfaction Problem. *Algorithms for Molecular Biology* 2012 **7**:15.

2.2.2 Finding steady states corresponding to steady fluxes

It is remarkable that because of the metabolic origin of SNA, many people do not really make any difference between steady fluxes and proper steady states. However, in the context of general intracellular biochemical models, the proper steady states, with a reasonable rate law, realize only a tiny fraction of the steady state flux cone.

Making such hypotheses on reasonable kinetics, [13] attempts to generalize the line of work developed for S-Systems by Savageau [56] and Voit [57], without resorting to the same power-law approximation.

[13] Faten Nabli and Sylvain Soliman. "Steady-state solution of biochemical systems, beyond S-Systems via T-invariants". In: *CMSB'10: Proceedings of the 8th International Conference on Computational Methods in Systems Biology*. Ed. by Paola Quaglia. CoSBi. Trento, Italy: ACM, Oct. 2010, pp. 14–22. ISBN: 978-1-4503-0068-1. DOI: 10.1145/1839764.1839768

Steady-state solution of biochemical systems, beyond S-Systems via T-invariants

Faten Nabli
Faten.Nabli@inria.fr

Sylvain Soliman*
Sylvain.Soliman@inria.fr

INRIA Paris-Rocquencourt, Équipe CONTRAINTES,
Domaine de Voluceau, Rocquencourt, BP 105,
78153 LE CHESNAY CEDEX - FRANCE

ABSTRACT

In recent years Systems Biology has become a rich field of study, trying to encompass all the information that has become available thanks to the new high-throughput techniques of biologists, in order to build detailed models of complex systems.

Some models have been growing bigger and bigger, but lacking most of precise kinetic data. Other models remain of reasonable size, but have an even larger uncertainty about parameter values. Unfortunately, very few analyses allow to extract information about the dynamics of these models when pure symbolic computations fails.

This article presents a way to generalize well-known results about the steady-state analysis of some symbolic Ordinary Differential Equations systems by taking into account the *structure* of the reaction network. The structural study of the underlying Petri net, usually used mostly for metabolic flux analysis, will provide classes where the computation of some steady states of the system is possible, even though the original symbolic model did not form an S-system and was not solvable by state-of-the-art symbolic computation software.

This new method is then illustrated on some models of the Biomodels repository and is followed by a brief discussion.

Categories and Subject Descriptors

D.2.2 [**Software engineering**]: Design Tools and Techniques—*Petri nets*; J.3 [**Computer applications**]: Life and medical sciences—*Biology and genetics*; D.1.6 [**Software**]: Logic Programming; I.6 [**Simulation and Modeling**]: Model Validation and Analysis

General Terms

Theory, Algorithms, Verification

*Corresponding author

CMSB '10, September 29 – October 1, Trento, Italy

Keywords

Systems biology, Steady-state analysis, S-systems, Mass Action, T-invariants

1. INTRODUCTION

In recent years Systems Biology has become a rich field of study, trying to encompass the huge amount of heterogeneous information that has become available thanks to the new high-throughput techniques of biologists, in order to make it usable through detailed models of complex systems.

Some models have been growing bigger and bigger, filled with more and more mechanistic details, especially recently acquired post-transcriptional information, but lacking most of precise kinetic data. Unfortunately, very few analyses allow to extract information about the dynamics of these models, either because of their size or of the imprecise kinetics.

Other models remain of reasonable size, but have an even larger uncertainty about parameter values. For this other kind of model it is also important to be able to provide some dynamical analysis of the system's behavior.

This article presents a way to generalize well-known results about the steady-state analysis of some symbolic Ordinary Differential Equations (ODE) systems by taking into account the *structure* of the reaction network. The structural study of the underlying Petri net (PN), usually used mostly for metabolic flux analysis, will provide classes where a symbolic reasoning close to that underpinning S-systems allows the computation of some steady states of the system. This property holds even though the original (symbolic) ODEs did not form an S-system and were not solvable by symbolic computation software like Maple[1].

After some preliminaries about Biochemical Systems Theory (BST) and Petri nets, Section 3 will describe how to combine invariant analysis, log-transform and Gaussian elimination to obtain some steady states. Section 4 will then show the application of that technique on some models of the Biomodels repository [17] and will be followed by a brief discussion.

The idea of applying BST to big biological networks is definitely not new [21, 29], however it is restricted to systems where the precise kinetics of the system, though unknown, can be approximated by an S-system. This approximation makes sense in several cases, notably gene-regulation maps,

[1]For experiments of Section 4, the most recent released version, i.e., Maple 13, was used.

however it does not cope with the cases where modellers have a precise kinetic expression in mind and only search for parameter values, which is quite frequent in mathematical models of small or medium size.

T-invariant analysis is also already frequently used, but it only provides *steady fluxes* and not steady states. The difference is discussed in details in Section 2.2. Note however that it can bring other informative content about the model (see for instance [11]).

Many other methods can be used to reason about the steady states of a partially unknown model, like bifurcation analysis (but it usually is restricted to very few parameters) or gene-regulation-network feedback-loop analysis [26], however they have a rather different range of application than the technique described in this paper.

2. PRELIMINARIES

In general, we will consider biochemical systems described by reactions (*à la* SBML[15]). More precisely the model is supposed to be formed by m reactions and dealing with n species; the j^{th} reaction has the following form:

$$\sum_{i=1}^{n} S_{ij} X_i \stackrel{e_j}{\Rightarrow} \sum_{i=1}^{n} S'_{ij} X_i$$

where S_{ij} and S'_{ij} are the stoichiometric coefficients (usually positive integers), and e_j is the rate function of the reaction.

2.1 Biochemical Systems Theory

In the late 1960s, Savageau introduced the Biochemical System Theory (BST) [21, 22] as a framework for modeling biochemical systems with ODEs.

The major advantage of this formalism is that it enables the modeller to describe the dynamics of a biochemical system knowing only the identity of reactants and their interconnections [28].

These ODEs have a canonical form using a power-law representation based on the General Mass Action (GMA) hypothesis. In the GMA form, rate functions are formulated as:

$$e_j = k_j \times \prod_i X_i^{S_{ij}}$$

where X_i is the concentration of the i^{th} species and k_j is the kinetic rate constant of the j^{th} reaction. This corresponds to the well-known and quite standard Mass Action kinetics derived from the concept of chemical equilibrium and representing intuitively the fact that reactions have a rate proportional to the amount of their reactants.

The change in the quantity of X_i is thus:

$$\begin{aligned}\frac{dX_i}{dt} &= \sum_{j=1}^{n} (S'_{ij} - S_{ij}) e_j \\ &= \sum_{j=1}^{n} S'_{ij} k_j \prod_i X_i^{S_{ij}} - \sum_{j=1}^{n} S_{ij} k_j \prod_i X_i^{S_{ij}}\end{aligned}$$

i.e., variation of a species per time is a difference between two sums of power-law functions, one associated to its production and the other to its consumption.

One special case of GMA is when every species is produced by at most one reaction and consumed through at most one reaction or when reactions producing each species are dependent and reactions consuming each species are dependent so that it becomes possible to combine the sum of power laws referring to the production term into a unique power law and combine the sum referring to the consumption term into one single power law.

Even when direct combination is not possible, aggregation through an approximation can be done, for instance close to some steady states. In that case the exponents in the power law become real numbers. Note that the applicability of such an approximation for highly non-linear systems when the steady states are yet to be determined is not always clear, which is one of the motivations for this very article.

Systems under this form are called S-systems where S refers to synergistic nature of this nonlinear form. In S-systems, the time rate of change of a species is written as:

$$\frac{dX_i}{dt} = k_i^+ \prod_{i=1}^{n} X_i^{\alpha_i} - k_i^- \prod_{i=1}^{n} X_i^{\beta_i}$$

where k_i^+ and k_i^- are positive rate constants.

One of the main properties of S-systems is that they can be analytically solved for steady states. Indeed being at steady state amounts to the fact that:

$$\forall i \quad \frac{dX_i}{dt} = 0 \tag{1}$$

which is equivalent to

$$\forall i \quad k_i^+ \prod_{i=1}^{n} X_i^{\alpha_i} = k_i^- \prod_{i=1}^{n} X_i^{\beta_i} \tag{2}$$

The system constituted of all equations (2) can be linearized by applying the logarithm function [22] and we obtain:

$$\forall i \quad \sum_{i=1}^{n} (\alpha_i - \beta_i) log(X_i) = log(\frac{k_i^-}{k_i^+}) \tag{3}$$

The system (3) can then be solved via standard Gaussian elimination.

This symbolic solution for steady states distinguishes S-systems from other non-linear ODE systems for which there is, in general, no simple solution. Actually, as will be detailed in Section 4, using the latest version of the Biomodels repository[2], almost half of the curated SBML models providing ODEs do not get any solution when imported into Maple for steady-state search in a reasonable time.

In Section 3 we will explain how to generalize this search for steady-state solution using T-invariants and apply it to Biomodels in Section 4.

2.2 Petri Nets

The use of Petri nets (PN) to represent biochemical reaction models, taking into account the difference between compounds and reactions in the graph, and make available various kinds of analyses is quite old [19], however it remains somehow focused towards mostly qualitative and structural properties, or metabolic flux analysis. PNs have recently been used for module decomposition, like (I/O) T-invariants [9, 11], which are related to the dynamical notion of elementary flux modes [23]. We will propose a novel use of T-invariants but first need to settle some basic notations.

[2]dated January 2010

A Petri net is a bipartite oriented multigraph of transitions, usually represented as square boxes, and places, usually represented as circles, that defines a transition relation on *markings* of the net, i.e., multisets of tokens associated to places. The relation is defined by *firings* of transitions, i.e., when there are tokens (as many as the weights of the incoming arcs) in all pre-places of a transition, they can be consumed and as many tokens as the weights on the outgoing arcs are added to each post-place.

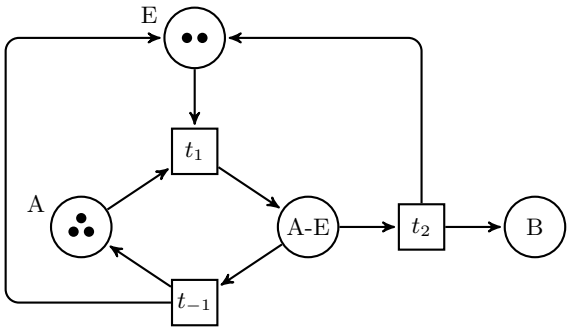


Figure 1: Biochemical model of Example 1, represented as a PN with a marking enabling t_1

EXAMPLE 1. *For instance the classical enzymatic reaction written*[3] *A + E <=>A−E =>B +E corresponds to the Petri net depicted in Figure 1.*

Reaction models can usually be easily represented as PNs by mapping compounds/species to places and reactions to transitions (stoichiometry corresponding to the weights of the arc between the places and the transition), see Figure 1.

P (resp. T) invariants are defined, as usual, as integer vectors V representing a multiset of places (resp. of transitions) such that $V \cdot I = 0$ (resp. $I \cdot V = 0$) where $I = I^+ - I^-$ is the *incidence matrix* of the Petri net, i.e. I_{ij} is the number of arcs from transition j to place i (I^+_{ij}), minus the number of arcs from place i to transition j (I^-_{ij}). Intuitively, a P-invariant is a multiset representing a weighting of the places and such that any such weighted marking remains invariant by any firing; a T-invariant represents a multiset of firings that will leave unchanged any marking.

Let us denote by $\mathcal{T}$ the (infinite) set of T-invariants. Since any positive linear combination of T-invariants is also a T-invariant, it is useful to consider the (finite) set of semi-positive minimal T-invariants $\mathcal{T}_{min}$ corresponding to T-invariants with minimal support and minimal value. $\mathcal{T}_{min}$ forms a basis of $\mathcal{T}$.

To compute the invariants of a Petri net, especially if this computation is combined with other Petri-net analyses, like sinks and sources, traps, deadlocks, etc. the most natural solution is to use a Petri-net dedicated tool like INA, PiNA, or Charlie for instance through the interface of Snoopy [12], which allows the import of SBML models as Petri nets. Standard integer methods like Fourier-Motzkin elimination will then provide an efficient means to compute P or T-invariants. These methods however generate lots of candidates which are afterwards eliminated and also need to incorporate some means (like equality class definition) to avoid combinatorial explosion at least in some simple cases.

Another way to extract the minimal semi-positive invariants of a model is to use one of the software tools that provide this computation for biological systems, generally as "conservation law" computation, and based on linear algebra methods like QR factorization [27]. This is the case for instance of the METATOOL [30] and COPASI [13] tools. The idea is to use a linear relaxation of the problem, which suits well very big graphs, but needs again *a posteriori* filtering of the candidate solutions.

We use Nicotine [24], a Constraint-Programming-based T and P-invariant Extractor. This tool[4] allows us import and export of several formats (APNN (a subset), PNML, SBML, BIOCHAM, etc.). Note however that the method presented is independent of any precise tool, as long as some T-invariants can be computed.

It is important to note that T-invariants are of course already related to steadyness. However, the cone of $\mathcal{T}$ defines the steady *fluxes* of the system, but does not relate to *states*, as defined by concentrations of compounds. Depending on the kinetic laws, some fluxes might be generated by some state while some others might not. For instance, if a T-invariant requires reaction 1 to fire twice as much as reaction 2 and both have the same kinetic expression, the flux does not correspond to any state. In this article we focus on steady states, as is classical in dynamical systems theory, and hence need to go one step further than T-invariants.

3. METHOD

Let us now consider a biochemical system described by reactions, as in Section 2, but without making any hypothesis on the form of the rate laws e_j except that it is *multiplicative*, i.e. $e_j(X) = 0 \Leftrightarrow \exists i, S_{ij} > 0, X_i = 0$. This condition is not very strong (GMA systems verify it but also Michaelian or Hill kinetics for instance) but is required in order to reason structurally on the fluxes of the system. It is quite common for structural or symbolic analyses (see for instance [8, 7, 16]) and is quite widely accepted by modellers and mathematical biologists. Note that it is strictly more general than GMA, which is often already a requirement for any stochastic simulation/analysis.

Finding a steady state amounts to solving:

$$\forall i \quad \frac{dX_i}{dt} = \sum_{j=1}^{n} (S'_{ij} - S_{ij}) e_j = 0$$

Considering the incidence matrix I of the Petri net corresponding to the biochemical model, the above system is equivalent to:

$$I \cdot E(X) = 0$$

where E is the vector $(e_1, \ldots, e_m)$.

This system is — in general — non-linear and cannot be analytically solved. Nevertheless, it is possible to try and solve systems corresponding to restricted cases using T-invariants, which will result in a correct but not complete method to obtain steady states. We will illustrate this method in the following, starting with a restricted version and then developing the technique in its full generality. The completeness of the method is discussed in Section 5.

[3] in BIOCHAM-like syntax [2]

[4] `http://contraintes.inria.fr/~soliman/nicotine`

As mentioned above, T-invariants are mathematically defined as positive vectors which are solutions of the equation $I \cdot V = 0$. Therefore, each steady state X defines an $E(X)$ which is a T-invariant. Conversely we have:

$$E(X) \in \mathcal{T} \Rightarrow \frac{dX}{dt} = 0$$

Finding the X solutions of the above equation is still intractable if one looks for any T-invariant, however one can state this sufficient condition in a different way:

Let V be a given T-invariant. Then solving the system $E(X) = V$ will lead to steady states of the original system.

3.1 Minimal T-invariants

Let us first consider a restricted case where:

1. $V = \alpha V', V' \in \mathcal{T}_{min}$, i.e., V is proportional to a minimal T-invariant;
2. V (equivalently V') has a GMA support, i.e.

$$\forall j \in support(V) \quad e_j = k_j \prod_{i=1}^{n} X_i^{S_{ij}}$$

Note that we only consider reaction networks with multiplicative kinetics, and that here we add the restriction that the kinetic laws of reactions in the support of the chosen T-invariant are in GMA form, thereby excluding Michaelian or other multiplicative kinetics from appearing in the support of V.

Solving $E(X) = V$ now amounts to solving (for X and α):

$$\begin{cases} (a) & k_j \prod_{i=1}^{n} X_i^{S_{ij}} = \alpha V'_j & j \in support(V') \\ (b) & \exists i, S_{ij} > 0, X_i = 0 & j \notin support(V') \end{cases} \quad (4)$$

Part (a) of system (4) is a direct consequence of our restricted setting, part (b) is a consequence of the *multiplicative* nature of the kinetics.

We will explain in the next section how to solve the second part, but let us first concentrate on the first part.

Even if we did not restrict ourselves to S-systems, it is now possible to log-linearize this subsystem:

$$\sum_{i=1}^{n} S_{ij} log(X_i) - log(\alpha) = log(\frac{V_j}{k_j}) \qquad j \in support(V') \quad (5)$$

We obtain $|support(V')|$ *linear* equations over at most $n+1$ unknowns and can then apply Gaussian elimination or any other equivalent method to obtain a (log-)vector space of steady states. Remark that it is almost immediate to deduce the matrix corresponding to system (5) from I^- the reactant part of the incidence matrix.

3.2 Solving for zeroes

Solving part (b) of system (4) is actually a simple matter of enumeration: one tries to nullify some reactions' rate by nullifying some concentrations. However one must also verify that all X_i involved in part (a) are strictly positive (otherwise there is no solution).

We implemented this as a simple enumerative search in Prolog, which is the language underlying the Nicotine tool and even for the biggest systems we tried the search for all solutions is under ten milliseconds. The only computationally expensive part being the search for T-invariants.

Remark that one could relax the restriction to multiplicative kinetics if one provides a generic way to solve (b) while ensuring the feasibility of (a). For instance one could consider that if a reaction has kinetic expression $e = k * (A + B) * C$ then $e = 0 \Leftrightarrow C = 0 \vee A = B = 0$. This kind of condition would be easy to incorporate into our scheme but needs to be formulated in a general way.

EXAMPLE 2. *Consider for instance the small model of the bacteriophage T7 of [1].*

It can be represented (in BIOCHAM syntax [2]) as 6 simple reactions (all Mass Action):

```
1 MA(c1) for          gen => tem.
2 MA(c2) for          tem => _.
3 MA(c3) for          tem => tem + gen.
4 MA(c4) for  gen + struc => virus.
5 MA(c5) for          tem => tem + struc.
6 MA(c6) for        struc => _.
```

The system has 2 semi-positive minimal T-invariants: [t1, t2, t3], [t5, t6], but solving for zeroes immediately leads to the fact that the only steady state is gen = tem = struc = 0.

3.3 Other T-invariants

The first version described above already works reasonably well at finding some steady states, however one of its shortcomings is that it only examines minimal T-invariants one by one, i.e., it restricts its search to the edges of the cone of steady fluxes.

The idea is to generalize the method while ensuring, when possible, that (4) remains solvable quite easily.

3.3.1 Disjoint support

Let us suppose that different minimal T-invariants $V'_1, \ldots, V'_k$ have disjoint GMA supports. One can now obtain a more general version of the method introduced in Section 3.1:

1. $V = \sum \alpha_i V'_i, V'_i \in \mathcal{T}_{min}$;
2. $\forall i \neq j \; support(V'_i) \cap support(V'_j) = \emptyset$;
3. $\forall j \in support(V) \quad e_j = k_j \prod_{i=1}^{n} X_i^{S_{ij}}$

Solving $E(X) = V$ now amounts to solving (for X and α):

$$\begin{cases} (a) & k_j \prod_{i=1}^{n} X_i^{S_{ij}} = \alpha_{j_0} V'_{j_0} & j \in support(V_{j_0}) \\ (b) & \exists i, S_{ij} > 0, X_i = 0 & j \notin support(V) \end{cases} \quad (6)$$

Since the supports are disjoint, there exists one and only one j_0 in part (a) of the system (4).

Now, simply notice that the new system is also log-linear and can thus be solved as before.

3.3.2 Closed support

It is possible to obtain an even more generalized version where more combinations of T-invariants are tested simply by trying to solve systems like (6) for some arbitrary minimal T-invariant combinations.

In the general case, the (a) part will become intractable (there is a sum at the right of the "=" sign); however it can sometimes be simplified, for instance when the obtained equation is actually linear (the S_{ij} on the left are equal to 0 or 1).

Since the computational cost of trying to solve the system (and stop if the symbolic computation fails) is very low

compared to T-invariant computation, many combinations can easily be tried. Trying all combinations remains however definitely impossible and to guide the search an idea is to notice that, in order to solve part (b) of the system while maintaining the solvability of (a), one needs to look for a T-invariant combination that is *closed*, in the sense of the Chemical Organization Theory (COT) [6].

Note, nevertheless, that it is not enough to look for T-invariant combinations whose support is a minimal organization, as in [3]: we should try all organizations, and might even prefer the biggest ones. Indeed, as is the case for T-invariants, restricting the search to minimal organizations will only lead to small subproblems when the complete system might be solvable. The objective should thus be to look for T-invariant combinations such that the combined support is an organization, is as big as possible, but such that the (a) part remains solvable.

The search procedure thus becomes: find all minimal T-invariants[5], look for minimal T-invariant combinations (starting with 0 or 1 invariant), try to solve the system, if (b) is not possible, add another minimal T-invariant such that it might close the support, *à la* [3], otherwise try to add any other minimal invariant.

EXAMPLE 3. *Consider again the bacteriophage T7 model of Example 2, but now let us forget about the virus, which will otherwise always increase; one obtains:*

```
MA(c1)  for            gen => tem.
MA(c2)  for            tem => _.
MA(c3)  for            tem => tem + gen.
MA(c4)  for   gen + struc => _.
MA(c5)  for            tem => tem + struc.
MA(c6)  for          struc => _.
```

rule 4 no longer includes the virus

Now the system gets a third minimal T-invariant: [t1, t2, t3], [t3, t4, t5], [t5, t6], but most importantly another steady state is found (it is the only non-trivial one actually), using a combination of all the minimal invariants (even if the supports are not disjoint):

$$\begin{aligned} tem &= \frac{c1 * c6 * (c2 - c3)}{c2 * c4 * (c3 - c2 - c5)} \\ struc &= \frac{c1 * (c3 - c2)}{c2 * c4} \\ gen &= \frac{c6 * (c2 - c3)}{c4 * (c3 - c2 - c5)} \end{aligned}$$

Note that other combinations of T-invariants, like MCT-sets [20], might be used, however they will still need to allow to solve (a) and (b), and thus to have as support an organization, hence the choice to remain at that level for now.

4. RESULTS

In this section, we apply the proposed method to the models of the Biomodels repository[6]. Of the 241 curated models, 14 do not include any continuous part (ODEs) and are thus not targetted by our method, for the other ones the structure and kinetic laws were extracted from the SBML by our tool, ignoring any other information (like events).

Using the Nicotine tool implementing the techniques described above, some steady states were found for 94 models out of the remaining 227, with a timeout of two minutes. Among the remaining 133 models, for which Nicotine could not find any solution, only 31 hit the time-out of two minutes. In most of the cases, the tested combinations of T-invariants do not lead to any steady state computation because of the strict conditions on the applicability of this method (namely multiplicative kinetics and GMA support, as explained in the previous section). As we will show below it is often possible to restructure an SBML model such that these conditions are met, however applying systematically such a procedure would be out of the scope of this article

This problem reflects the fact that modellers often tend to combine several separate reactions into one, leading to a wrong structure (and non-multiplicative kinetics). For instance, in model 149 (actually `BIOMD0000000149.xml`, but for the sake of simplicity we will from now on skip the prefix) of ERK crosstalk, the "Axin synthesis" reaction R14 is given as a single reaction X11 +X14 =>X11 +X14 +X12, i.e., a synthesis of X12 with two modifiers, X11 and X14 and a single rate law: $v_{14} = k_{14} + k_{21} * (X11 + X14)$.

Splitting that reaction into:

```
1 MA(k14)  for  _      =>          X12.
2 MA(k21)  for  _  =[X11]=>        X12.
3 MA(k21)  for  _  =[X14]=>        X12.
4
5
6
```

would have allowed a multiplicative kinetics with the same ODEs but a differently structured model. The question of *properly structured* SBML models is discussed in more details in the conclusion.

Note that trying to solve analytically all the models, even with state-of-the-art symbolic computation software like Maple 13, does not solve all problems either. Indeed, of the 227 curated SBML models with an ODE part (automatically generated in Maple format from our tool and shell scripts), 105 models do not provide any solution when imported into Maple for steady-state search, with a timeout of two minutes.

Interestingly, Nicotine finds some steady states for 31 models of the 105 Maple failures to give any steady-state solution even though there are ODEs.

Consider for instance model 46, describing the mechanism of protection of peroxidase activity by oscillatory dynamics It contains 16 places (chemical species) and 15 transitions (chemical reactions). All kinetics laws are multiplicative, the method of Section 3 thus applies and we obtain some steady states in 28ms[7], even if Maple did not provide any result.

The minimal T-invariant [v131, v132] gives four families of steady states represented in the following table, and corresponding to the different ways to solve for zeroes:

[5] As discussed in Section 5 the method can actually be applied even if only some invariants have been found.

[6] dated January 2010

[7] computation time on a PC with an intel Core2 Quad processor 2.8GHz and 8Go of memory. We used the same PC for the whole procedure, including T-invariant computation

species	family1	family2	family3	family4
NADH	0	0	0	0
O2	$*>0$	$*>0$	$*>0$	$*>0$
H2O2	*	*	0	0
per3	0	0	*	*
coI	*	0	*	0
ArH	0	*	0	*
coII	*	0	*	0
Ar	*	*	*	*
NADrad	0	0	0	0
super	0	0	0	0
coIII	*	*	*	*
per2	0	0	0	0
NAD2	*	*	*	*
NAD	*	*	*	*
O2g	$*>0$	$*>0$	$*>0$	$*>0$
NADHres	0	0	0	0

In this table the symbol (*) denotes that any concentration value (positive or null) is acceptable. It shows that, to the considered minimal T-invariant corresponds an infinity of steady states where some elements may have non zero concentration values.

For the non-null values, Gaussian elimination solves the system of linear equations. In this example, an equilibrium is established between O2 and O2g:

$$\frac{\mathrm{O2}}{\mathrm{O2g}} = \frac{k13f}{k13b}$$

where $k13f$ and $k13b$ are respectively the rate constants of reactions v131 and v132.

Note also that some steady states can be extracted even from the trivial T-invariant: 8 families of concentrations are computed, in all of them the concentrations of O2, NADrad, super, O2g and NADres is null and the concentration of coIII, NAD2 and NAD have positive or null values.

Some more complex steady states can also be found as is the case for model number 9 of the Biomodels database, describing the classical Huang and Ferrell model of the MAPK cascade [14] (see Figure 2). Note that since the structure of this model was used as a basis for [18], it has already been studied in detail in [9].

This model contains 25 places and 30 transitions; steady states computation is fulfilled in 220 ms and reveals 15 minimal T-invariants.

The family of T-invariants [[r1a, r1arev], [r2a, r2arev], [r2a, r2b, r1a, r1b]], which denotes a linear combination of the three minimal T-invariants [r1a, r1arev], [r2a, r2arev] and [r2a,r2b,r1a,r1b], gives four families of steady states represented in the following table:

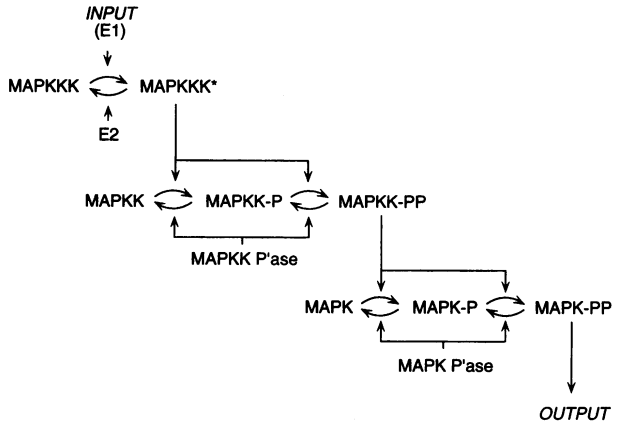


Figure 2: Figure 1 of [14] describing, in a diagram, model 9 of the Biomodels repository. In the SBML model and thus in our results, the names are changed from "MAPK-P" to "P_K" (i.e., drop "MAP" and put phosphorylations first).

species	family1	family2	family3	family4
E1	$*>0$	$*>0$	$*>0$	$*>0$
E2	$*>0$	$*>0$	$*>0$	$*>0$
KKK	$*>0$	$*>0$	$*>0$	$*>0$
P_KKK	$*>0$	$*>0$	$*>0$	$*>0$
KK	0	0	0	0
P_KK	0	0	0	0
PP_KK	0	0	*	*
K	*	*	0	0
P_K	0	*	0	0
PP_K	0	*	0	*
KPase	*	0	*	0
KKPase	*	*	0	0
E1_KKK	$*>0$	$*>0$	$*>0$	$*>0$
E2_P_KKK	$*>0$	$*>0$	$*>0$	$*>0$
P_KKK_KK	0	0	0	0
P_KKK_P_KK	0	0	0	0
PP_KK_K	0	0	0	0
PP_KK_P_K	0	0	0	0
KKPase_PP_KK	0	0	0	0
KKPase_P_KK	0	0	0	0
KPase_PP_K	0	0	0	0
KPase_P_K	0	0	0	0
K_PP_norm	*	*	*	*
KK_PP_norm	*	*	*	*
KKK_P_norm	*	*	*	*

For each of these families, we obtain a steady state when the non null values in the table, E1, E2, KKK, P_KKK, E1_KKK and E2_P_KKK satisfy the following equations:

$$\begin{aligned} \frac{\mathrm{E1} * \mathrm{KKK}}{\mathrm{E1\_KKK}} &= \frac{k2+d1}{a1} \\ \mathrm{E1\_KKK} &= \mathrm{E2\_P\_KKK} \\ \frac{\mathrm{E2} * \mathrm{P\_KKK}}{\mathrm{E2\_P\_KKK}} &= \frac{d2+k2}{a2} \end{aligned}$$

where $k2, d1, a1$ and $a2$ are the rate constants of r1b (and r2b), r1arev, r1a and r2a respectively.

This can be summed up by stating that if most of the

complexes involved in the cascade are absent, most notably those involving phosphatases and the second-level kinase, there exists a steady state where the first and third level maintain some equilibrium.

A second set of steady states results from the T-invariant combination [[r3a, r3arev], [r4a, r4arev], [r4a, r4b, r3a, r3b], [r5a, r5arev], [r6a, r6arev], [r6a, r6b, r5a, r5b]] and defines again four families of steady states as follows:

species	family1	family2	family3	family4
E1	*	*	0	0
E2	0	0	0	0
KKK	0	0	*	*
P_KKK	$*>0$	$*>0$	$*>0$	$*>0$
KK	$*>0$	$*>0$	$*>0$	$*>0$
P_KK	$*>0$	$*>0$	$*>0$	$*>0$
PP_KK	$*>0$	$*>0$	$*>0$	$*>0$
K	0	0	0	0
P_K	0	0	0	0
PP_K	0	*	0	*
KPase	*	0	*	0
KKPase	$*>0$	$*>0$	$*>0$	$*>0$
E1_KKK	0	0	0	0
E2_P_KKK	0	0	0	0
P_KKK_KK	$*>0$	$*>0$	$*>0$	$*>0$
P_KKK_P_KK	$*>0$	$*>0$	$*>0$	$*>0$
PP_KK_K	0	0	0	0
PP_KK_P_K	0	0	0	0
KKPase_PP_KK	$*>0$	$*>0$	$*>0$	$*>0$
KKPase_P_KK	$*>0$	$*>0$	$*>0$	$*>0$
KPase_PP_K	0	0	0	0
KPase_P_K	0	0	0	0
K_PP_norm	*	*	*	*
KK_PP_norm	*	*	*	*
KKK_P_norm	*	*	*	*

The additional equations are:

$$\begin{aligned}
\frac{\text{KK} * \text{P_KKK}}{\text{P_KKK_KK}} &= \frac{k3+d3}{a3} \\
\frac{\text{KKPase} * \text{P_KK}}{\text{KKPase_P_KK}} &= \frac{d4+k4}{a4} \\
\frac{\text{KKPase_PP_KK}}{\text{P_KK} * \text{P_KKK}} &= \frac{a5*k5}{k6*(d5+k5)} \\
\frac{\text{KKPase_P_KK}}{\text{P_KKK_KK}} &= \frac{k3}{k4} \\
\frac{\text{P_KK} * \text{P_KKK}}{\text{P_KKK_P_KK}} &= \frac{k5+d5}{a5}
\end{aligned}$$

$$\frac{\text{PP\_KK} * \text{P\_KKK\_KK}}{P\_KK^2 * P\_KKK} = \frac{a5*k5*(d6+k6)*a4*k4}{k3*k6*(d4+k4)*a6*(k5+d5)}$$

This complex equilibrium (dimension 9, ignoring the variables completely free in the above table) describes a steady state where the last level is off (K = 0) but the first two are actually active, especially the intermediary level.

In the same way, steady states are found for the T-invariant combination [[r10a, r10arev], [r10a, r10b, r9a, r9b], [r7a, r7arev], [r8a, r8arev], [r8a, r8b, r7a, r7b], [r9a, r9arev]]:

species	family1	family2	family3	family4
E1	*	*	0	0
E2	*	0	*	0
KKK	0	0	*	*
P_KKK	0	*	0	*
KK	*	0	*	0
P_KK	*	0	*	0
PP_KK	$*>0$	$*>0$	$*>0$	$*>0$
K	$*>0$	$*>0$	$*>0$	$*>0$
P_K	$*>0$	$*>0$	$*>0$	$*>0$
PP_K	$*>0$	$*>0$	$*>0$	$*>0$
KPase	$*>0$	$*>0$	$*>0$	$*>0$
KKPase	0	0	0	0
E1_KKK	0	0	0	0
E2_P_KKK	0	0	0	0
P_KKK_KK	0	0	0	0
P_KKK_P_KK	0	0	0	0
PP_KK_K	$*>0$	$*>0$	$*>0$	$*>0$
PP_KK_P_K	$*>0$	$*>0$	$*>0$	$*>0$
KKPase_PP_KK	0	0	0	0
KKPase_P_KK	0	0	0	0
KPase_PP_K	$*>0$	$*>0$	$*>0$	$*>0$
KPase_P_K	$*>0$	$*>0$	$*>0$	$*>0$
K_PP_norm	*	*	*	*
KK_PP_norm	*	*	*	*
KKK_P_norm	*	*	*	*

and some complex equilibrium between PP_KK, K, P_K, PP_K, KPase, PP_KK_K , PP_KK_P_K, KPase_PP_K, and KPase_P_K is defined by the following equations:

$$\begin{aligned}
\frac{\text{K} * \text{PP_KK}}{\text{PP_KK_K}} &= \frac{k7+d7}{a7} \\
\frac{\text{KPase} * \text{P_K}}{\text{KPase_P_K}} &= \frac{d8+k8}{a8} \\
\frac{\text{KPase_PP_K}}{\text{PP_KK_P_K}} &= \frac{k9}{k10} \\
\frac{\text{KPase_P_K}}{\text{PP_KK_K}} &= \frac{k7}{k8} \\
\frac{\text{PP_K} * \text{PP_KK_K}}{\text{PP_KK_P_K} * \text{P_K}} &= \frac{k8*k9*a8*(k10+d10)}{k7*k10*(d8+k8)*a10} \\
\frac{\text{PP_KK} * \text{P_K}}{\text{PP_KK_P_K}} &= \frac{d9+k9}{a9}
\end{aligned}$$

This shows that the two last levels can also remain in a complex equilibrium.

Finally, note that for this model too the trivial T-invariant leads to some steady states.

5. CONCLUSION

We have presented a new method to compute, in a fully analytical way, steady states of biochemical systems defined by a system of ODEs. This technique generalizes the S-systems' log transformation to linearize equations obtained from T-invariants and thus leverages the information from the structure of the models, i.e., the underlying Petri net. This result, which becomes crucial when precise kinetic parameter values are not available, allows us to obtain analytical solutions for some steady states in about 30% of the cases where state-of-the-art symbolic computation led to a dead end, as demonstrated on the Biomodels database.

The first step of our method relies on the computation of the T-invariants of the model's Petri net. This, however

introduces two limitations:

1. actually, most of the computation time is spent on this first stage. Indeed, T-invariant computation is a hard problem (actually quite harder in practice than P-invariant computation on biochemical systems, though they have the same theoretical complexity. This seems related to the fact that in biochemical reaction networks the degree of the transitions is usually quite smaller than that of the places).

2. the T-invariants, and even the following step of solving system (6) suppose that the structure of the PN is *coherent* with the kinetics. Namely we require multiplicative kinetics for the whole model (and GMA for the support of some T-invariants).

Even if T-invariant computation gets more and more efficient (see for instance [4] for some recent work using decision diagrams), it is quite crucial to note that when T-invariant computation reveals time consuming, the proposed technique can work as soon as some (minimal, or even just candidates-minimal) T-invariants are found. Two examples that come to mind are thus the use of Nicotine [24] to compute T-invariants with an increasing bound on the integer domain, or to use [5] to compute the K-shortest minimal T-invariants, before (or while) proceeding to invariant composition and solving.

Addressing the other limitation is actually a much more general question. Indeed more and more formal techniques extract qualitative information from the structure of biochemical models, however the current status of hand-written models in web-based repositories is that the structure might be quite different from what the original modellers had in mind (or on diagram), even if the models are "curated". This issue applies to model-checking, abstract interpretation from the structure [7], stochastic simulation *à la* Gillespie [10], Chemical Reaction Network Theory [8, 25], etc. Some authors have already proposed solutions to check if the kinetics and the structure were at least coherent in some sense, notably [16] in order to use COT. Proposals allowing to obtain a *properly structured* model, as was done in the beginning of Section 4 are also under way (an article has recently been submitted about this topic by the second author). If everything else fails, it remains possible, as outlined in Section 3.2, to provide constraints corresponding to the non-multiplicative kinetics of the model and allowing to reason on the nullity of the kinetic expression.

Another noteworthy remark about the proposed method is that it is in general incomplete, since combinations involving many T-invariants usually result in a non-log-linearizable system. However, the method can be complete under certain conditions. For instance, for both versions of Example 2, the steady states that are found are the only ones, and the method can certify this fact. In this specific case the proof is quite simple: all combinations of minimal T-invariants are tried and either lead to no solution (when solving for zeroes) or are solvable. Finding more general conditions under which the described technique ensures that all possible steady states were found is one of our current perspectives.

Acknowledgements

This work has been partly supported by ANR Calamar (ANR-08-SYSC-003) and OSEO R&D Program BioIntelligence.

We also thank the referees for their extremely useful suggestions.

6. REFERENCES

[1] A. Alfonsi, E. Cancès, G. Turinici, B. di Ventura, and W. Huisinga. Adaptive simulation of hybrid stochastic and deterministic models for biochemical systems. *ESAIM: Proc.*, 14:1–13, Sept. 2005.

[2] L. Calzone, F. Fages, and S. Soliman. BIOCHAM: An environment for modeling biological systems and formalizing experimental knowledge. *Bioinformatics*, 22(14):1805–1807, 2006.

[3] F. Centler, C. Kaleta, P. S. di Fenizio, and P. Dittrich. Computing chemical organizations in biological networks. *Bioinformatics*, 24(14):1611–1618, May 2008.

[4] G. Ciardo, G. Mecham, E. P. Adet, and M. Wan. P-semiflow computation with decision diagrams. In *PETRI NETS '09: Proceedings of the 30th International Conference on Applications and Theory of Petri Nets*, pages 143–162, Berlin, Heidelberg, 2009. Springer-Verlag.

[5] L. F. de Figueiredo, A. Podhorski, A. Rubio, C. Kaleta, J. E. Beasley, S. Schuster, and F. J. Planes. Computing the shortest elementary flux modes in genome-scale metabolic networks. *Bioinformatics*, 25(23):3158–3165, Dec. 2009.

[6] P. Dittrich and P. di Fenizio. Chemical organisation theory. *Bulletin of Mathematical Biology*, 69(4):1199–1231, Apr. 2007.

[7] F. Fages and S. Soliman. Abstract interpretation and types for systems biology. *Theoretical Computer Science*, 403(1):52–70, 2008.

[8] M. Feinberg. Mathematical aspects of mass action kinetics. In L. Lapidus and N. R. Amundson, editors, *Chemical Reactor Theory: A Review*, chapter 1, pages 1–78. Prentice-Hall, 1977.

[9] D. Gilbert, M. Heiner, and S. Lehrack. A unifying framework for modelling and analysing biochemical pathways using petri nets. In *CMSB'07: Proceedings of the fifth international conference on Computational Methods in Systems Biology*, volume 4695 of *Lecture Notes in Computer Science*. Springer-Verlag, 2007.

[10] D. T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *Journal of Physical Chemistry*, 81(25):2340–2361, 1977.

[11] E. Grafahrend-Belau, F. Schreiber, M. Heiner, A. Sackmann, B. H. Junker, S. Grunwald, A. Speer, K. Winder, and I. Koch. Modularization of biochemical networks based on a classification of petri net by T-invariants. *BMC Bioinformatics*, 9(90), Feb. 2008.

[12] M. Heiner, R. Richter, and M. Schwarick. Snoopy - a tool to design and animate/simulate graph-based formalisms. In *Procedings of the International Workshop on Petri Nets Tools and APplications (PNTAP 2008)*, Marseille, Mar. 2008. ACM Digital Library. to appear.

[13] S. Hoops, S. Sahle, R. Gauges, C. Lee, J. Pahle, N. Simus, M. Singhal, L. Xu, P. Mendes, and U. Kummer. Copasi – a complex pathway simulator. *BioInformatics*, 22(24):3067–3074, 2006.

[14] C.-Y. Huang and J. E. Ferrell, Jr. Ultrasensitivity in the mitogen-activated protein kinase cascade. *PNAS*, 93(19):10078–10083, Sept. 1996.

[15] M. Hucka et al. The systems biology markup language (SBML): A medium for representation and exchange of biochemical network models. *Bioinformatics*, 19(4):524–531, 2003.

[16] C. Kaleta, S. Richter, and P. Dittrich. Using chemical organization theory for model checking. *Bioinformatics*, 25(15):1915–1922, 2009.

[17] N. le Novère et al. BioModels Database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems. *Nucleic Acid Research*, 1(34):D689–D691, Jan. 2006.

[18] A. Levchenko, J. Bruck, and P. W. Sternberg. Scaffold proteins may biphasically affect the levels of mitogen-activated protein kinase signaling and reduce its threshold properties. *PNAS*, 97(11):5818–5823, May 2000.

[19] V. N. Reddy, M. L. Mavrovouniotis, and M. N. Liebman. Petri net representations in metabolic pathways. In L. Hunter, D. B. Searls, and J. W. Shavlik, editors, *Proceedings of the 1st International Conference on Intelligent Systems for Molecular Biology (ISMB)*, pages 328–336. AAAI Press, 1993.

[20] A. Sackmann, M. Heiner, and I. Koch. Application of petri net based analysis techniques to signal transduction pathways. *BMC Bioinformatics*, 7(482), Nov. 2006.

[21] M. A. Savageau. Biochemical systems analysis i some mathematical properties of the rate law for the component enzymatic reactions. *Journal of Theoretical Biology*, 25(3):365–369, 1969.

[22] M. A. Savageau. Biochemical systems analysis ii the steady-state solutions for an n-pool system using a power-law approximation. *Journal of Theoretical Biology*, 25(3):370–379, 1969.

[23] S. Schuster, D. A. Fell, and T. Dandekar. A general definition of metabolic pathways useful for systematic organization and analysis of complex metabolic networks. *Nature Biotechnology*, 18:326–332, 2002.

[24] S. Soliman. Modelling biochemical reaction networks with biocham extracting qualitative and quantitative information from the structure. In *Proceedings of the 6th Vienna Conference on Mathematical Modelling MATHMOD'09*, volume 35, pages 2304–2312. ARGESIM, Feb. 2009.

[25] N. Soranzo and C. Altafini. ERNEST: a toolbox for chemical reaction network theory. *Bioinformatics*, 2009. In press (advance access).

[26] R. Thomas. On the relation between the logical structure of systems and their ability to generate multiple steady states or sustained oscillations. *Springer Ser. Synergetics*, 9:180–193, 1981.

[27] R. R. Vallabhajosyulaa, V. Chickarmane, and H. M. Sauro. Conservation analysis of large biochemical networks. *BioInformatics*, Nov. 2005. Advance Access.

[28] E. O. Voit. *Computational Analysis of Biochemical Systems. A Practical Guide for Biochemists and Molecular Biologists*. Cambridge University Press, Cambridge, U.K., 2000.

[29] E. O. Voit and T. Radivoyevitch. Biochemical systems analysis of genome-wide expression data. *Bioinformatics*, 16(11):1023–1037, 2000.

[30] A. von Kamp and S. Schuster. Metatool 5.0: fast and flexible elementary modes analysis. *Bioinformatics*, 22(15):1930–1931, 2006.

2.3 Siphons and Traps

Siphons and traps are the boolean counterparts of P/T-invariants. As explained above, they have been studied in the modeling of biochemical systems to detect pools of species that disappear or never do. They are also interesting in a more general way through notably the Siphon-Trap Property (STP) that holds when every siphon of a net contains a marked trap, and guarantees deadlock-freeness (no steady state).

In their study it was interesting both to use similar techniques as in Section 2.2.1, but also to compare our approach to the state of the art in boolean constraint satisfaction, namely SAT-solvers.

The result obtained is both more efficient than the known techniques for siphons/-traps computation, but also interesting in that SAT and CP appear to be comparable for this specific enumeration problem. This relies on an ad-hoc replay strategy for CP that can be generalized to many other enumeration problems.

An interesting outcome of this article is also the clarification of complexity classes for STP and other siphon/trap-related properties in the general and bounded-tree-width cases. This might give some clues for the practical performance observed on biochemical models.

[12] Faten Nabli, Thierry Martinez, François Fages, and Sylvain Soliman. "On Enumerating Minimal Siphons in Petri nets using CLP and SAT solvers: Theoretical and Practical Complexity". In: *Constraints* 21.2 (2016), pp. 251–276. ISSN: 1383-7133. DOI: 10.1007/s10601-015-9190-1

Noname manuscript No.
(will be inserted by the editor)

On Enumerating Minimal Siphons in Petri nets using CLP and SAT solvers: Theoretical and Practical Complexity*

Faten Nabli · Thierry Martinez · Franois Fages · Sylvain Soliman

Received: date / Accepted: date

Abstract Petri nets are a simple formalism for modeling concurrent computation. They are also an interesting tool for modeling and analysing biochemical reaction systems, bridging the gap between purely qualitative and quantitative models. Biological networks can indeed be complex, large, and with many unknown kinetic parameters, which makes the development of quantitative models difficult. In this paper, we focus on the Petri net representation of biochemical reactions and on two structural properties of Petri nets, siphons and traps, that bring us information about the persistence of some molecular species, independently of the kinetics. We first study the theoretical time complexity of minimal siphon decision problems in general Petri nets, and present three new complexity results: first, we show that the existence of a siphon of a given cardinality is NP-complete; second, we prove that deciding the Siphon-Trap property is co-NP-complete; third, we prove that deciding the existence of a minimal siphon containing a given set of places, deciding the existence of a siphon of a given cardinality and deciding the Siphon-Trap property can be done in linear time in Petri nets of bounded tree-width. Then, we present a Boolean model of siphons and traps, and two methods for enumerating all minimal siphons and traps of a Petri net, by using a SAT solver and a Constraint Logic Program (CLP) respectively. On a benchmark of 345 Petri nets of hundreds of places and transitions, extracted from biological models from the BioModels repository, as well as on a benchmark composed of 80 Petri nets from the Petriweb database of industrial processes, we show that both the SAT and CLP methods are overall faster by one or two orders of magnitude compared to the state-of-the-art algorithm from the Petri net community, and are in fact able to solve all the enumeration problems of our practical benchmarks.

* This paper is an extended version of a paper presented at CP 2012 [37].

EPI Contraintes, Inria Paris-Rocquencourt, France
Tel.: +33 1 39 63 56 37
Fax: +33 1 39 63 54 69
E-mail: {Thierry.Martinez, Francois.Fages, Sylvain.Soliman}@inria.fr, faten7@gmail.com

We investigate why these programs perform so well in practice, and provide some elements of explanation related to our theoretical complexity results.

Keywords Petri nets · Siphons · Traps · Systems Biology · SAT · Constraint Logic Programming

1 Introduction

Petri nets were introduced in the 60's as a simple formalism for describing and analyzing information processing systems that are characterized as being concurrent, asynchronous, non-deterministic and possibly distributed [40]. The use of Petri nets for representing biochemical reaction systems, by mapping molecular species to places and reactions to transitions, was introduced quite late in [41], together with some Petri net concepts and tools for the analysis of metabolic networks. In particular, the traditional Petri net concepts of place-invariants (P-invariants) and transition-invariants (T-invariants) have important interpretations in biochemical networks: P-invariants express structural conservation laws between molecular species, which correspond to linear invariants and possible variable eliminations in systems of ordinary differential equations, while T-invariants revealed to be equivalent to the notion of extremal fluxes in metabolic networks [49], one of the main tools for analyzing and optimizing metabolic networks [30, 15, 47]. Constraint logic programs have been proposed to compute P-invariants and T-invariants in [44] and [38] respectively. Constraint programming methods have also been applied successfully to many other biology related problems. For instance by Devloo et al. to discover efficiently the steady-state of large gene regulation networks [16]. Fanchon and al. use constraints to infer ranges of parameter values from observations [20] and for analysing discrete genetic regulatory networks [8]. Bockmayr and Courtois [3] use Hybrid Concurrent Constraint to model a variety of biological phenomena, such as reaching thresholds, kinetics, gene interaction or biological pathways. In [30], Larhlimi and Bockmayr take advantage of the implicit representation that constraints bring, to describe the elementary flux cone of some metabolic pathways. Backofen et al. pioneered in [1] the use of constraints and symmetry breaking for predicting the structure of proteins, etc.

In this paper, we consider the Petri net concepts of siphons and traps. A siphon is a set of places that, once unmarked, remains unmarked. A trap is a set of places that, once marked, can never loose all its tokens. Thus, siphons and traps have opposed effects on the token distribution in a Petri net. These structural properties provide sufficient conditions for reachability (whether the system can produce a given protein or reach a given state from a given initial state) and liveness (deadlock freedom from a given initial state) properties in ordinary Petri nets. It is proved that in order for a net to have all its transitions live, it is necessary that each siphon remains marked. Otherwise (i.e., once a siphon is empty), transitions having their input places in a siphon cannot be live. One way to keep each siphon marked is to have a marked trap inside it.

In fact, this condition is necessary and sufficient for a free-choice net to be live [40],

We first study the theoretical time complexity of siphon extraction problems in general Petri nets. It has been shown in [46] and [48] that the problems of existence of a minimal siphon of a given cardinality (k-MinimalSiphon), or containing a given place (Q-MinimalSiphon), are NP-complete, and recently in [39] that the siphon-trap property (stp) is in co-NP. Here we provide new theoretical complexity results. First we show the NP-completeness of the existence problem of a siphon of a given cardinality (k-Siphon) and thus the NP-hardness of MinimalCardinalitySiphon. Second, we prove that deciding the siphon-trap property is in fact co-NP-complete. Third, we prove that deciding the existence of a minimal siphon containing a given set of places, deciding the existence of a siphon of a given cardinality and deciding the Siphon-Trap property are of linear time complexity in Petri nets with bounded tree-width. These latter results follow from Courcelle's theorem.

Then we consider a simple Boolean model for defining siphons and traps, and two methods for enumerating the set of all minimal siphons and traps of a Petri net. The first method iterates the resolution of the Boolean model executed with a SAT solver, while the second proceeds by backtracking with a Constraint Logic Program over Booleans (CLP(B)).

We compare this Boolean constraint solving approach to the state-of-the-art algorithms from the Petri net community described in [11] for computing minimal sets of siphons and traps, which have already been shown to outperform Mixed Integer Linear Programs previously proposed in [35,9]. On a benchmark composed of 345 curated biological models of hundreds of species and reactions each, from the BioModels [1] repository [32], and of 80 Petri nets from the Petriweb [2] [22] database of industrial processes, we show that the SAT solver MiniSAT and CLP(B) solver GNU-Prolog are both faster by one or two orders of magnitude than the dedicated algorithms, and can in fact enumerate all solutions for all the intances of those benchmarks in a seconds. Finally, we question ourselves why these programs perform so well in practice, and provide some elements of explanation related to our theoretical complexity results.

2 Preliminaries on Petri Nets

2.1 Petri Nets

Definition 1 A *Petri net graph* N is a weighted bipartite directed graph $N = (P, T, W)$, where P is a non-empty finite set of vertices called *places*, T is a non-empty finite set of vertices called *transitions*, $P \cap T = \emptyset$, and $W : (P \times T) \cup (T \times P) \to \mathbb{N}$ is a weight function attached to the arcs.

[1] http://www.biomodels.net/

[2] http://www.petriweb.org/

Intuitively, the weight of a $P \times T$ arc represents the minimum number of tokens (molecules) necessary to enable a transition and the weight on a $T \times P$ arc represents the quantity produced. By abuse of notation, the weight is equal to one if it is omitted. The weight of a $P \times T$ (resp. $T \times P$) arc equals zero if there exist no arcs from place to transition (resp. from transition to place).

Definition 2 A *marking* for a Petri net graph is a mapping $m : P \to \mathbb{N}$ which assigns a number of tokens to each place. We say that a place p is *marked* if $m(p) > 0$, otherwise it is said to be *unmarked*.

Definition 3 A *Petri net* is a 4-tuple (P, T, W, m_0) where (P, T, W) is a Petri net graph and m_0 is an *initial marking*.

Let $N = (P, T, W)$ be a Petri net graph.

Definition 4 The set of *predecessors* (resp. *successors*) of a transition $t \in T$ is the set of places ${}^\bullet t = \{p \in P \mid W(p,t) > 0\}$ (resp. $t^\bullet = \{p \in P \mid W(t,p) > 0\}$). Similarly, the set of predecessors (resp. successors) of a place $p \in P$ is the set of transitions ${}^\bullet p = \{t \in T \mid W(t,p) > 0\}$ (resp. $p^\bullet = \{t \in T \mid W(p,t) > 0\}$).

The set of predecessors (resp. successors) ${}^\bullet S$ (resp. $S^\bullet$) of a set of places S is the union of sets of predecessors (resp. successors) of each place $p \in S$: ${}^\bullet S = \bigcup_{p \in S} {}^\bullet p$ (resp. $S^\bullet = \bigcup_{p \in S} p^\bullet$).

The set of predecessors (resp. successors) ${}^\bullet Q$ (resp. $Q^\bullet$) of a set of transitions Q is the union of sets of predecessors (resp. successors) of each transition $t \in Q$: ${}^\bullet Q = \bigcup_{t \in Q} {}^\bullet t$ (resp. $Q^\bullet = \bigcup_{t \in Q} t^\bullet$).

Definition 5 N is *ordinary* if for all $p \in P$ and for all $t \in T$, $W(p,t) \leq 1$ and $W(t,p) \leq 1$.

Definition 6 A transition t is *enabled* at marking m when $\forall p \in {}^\bullet t : m(p) \geq W(p,t)$.

For every two markings $m, m' : P \to \mathbb{N}$ and every transition $t \in T$, there is a *transition step* $m \xrightarrow{t} m'$ if for all $p \in P$, $m(p) \geq W(p,t)$ and $m'(p) = m(p) - W(p,t) + W(t,p)$.

$m \xrightarrow{t} m'$ means that the transition t is enabled in m and its firing leads to m'. An enabled transition may or may not fire if there are other transitions enabled.

Example 1 The classical Petri net view of a reaction model is to associate biochemical *species* to *places* and biochemical *reactions* to *transitions*. The well-known system of Michaelis-Menten enzymatic reactions can be represented by the Petri net depicted in Figure 1. It consists of three reactions that take place in two discrete steps: the first involves the reversible formation of a complex (AE) between the enzyme (E) and its substrate (A), and the second step involves an irreversible transformation of the product (B) with release of the enzyme.

$$A + E \leftrightarrows AE \to B + E$$

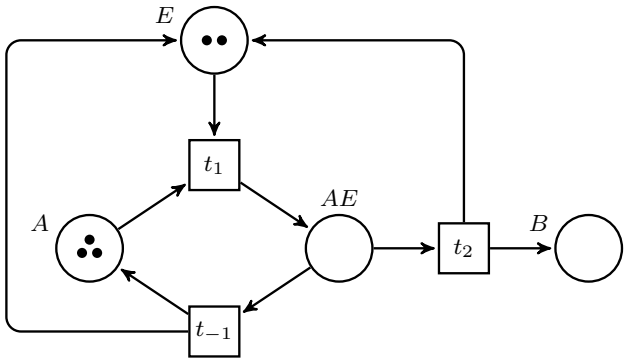


Fig. 1: Petri net associated to the example of Michaelis-Menten of Example 1, displayed here with a marking enabling transition t_1.

Definition 7 Given m and m' two markings of N. A finite sequence of transitions $\sigma = (t_0 \dots t_n)$ is a finite *firing sequence* of the Petri net if there exists a sequence of markings $m_1, \dots, m_n$ for which $m \overset{t_0}{\to} m_1 \overset{t_1}{\to} \dots \overset{t_{n-1}}{\to} m_n \overset{t_n}{\to} m'$. This is denoted by $m \overset{\sigma}{\to} m'$.

Definition 8 A marking m' is *reachable* from m if there exists a finite sequence σ such that $m \overset{\sigma}{\to} m'$.

Definition 9 Let $N = (P, T, W, m_0)$ be a Petri net.

- A transition $t \in T$ is *dead* at marking m if it is not enabled in any marking m' reachable from m.
- A marking m is *dead* if there is no transition enabled in m.
- A Petri net is *deadlock free* (weakly live) if there is no reachable dead markings.

2.2 Siphons and Traps

Let $N = (P, T, W)$ be a Petri net graph.

Definition 10 A *trap* is a non-empty set of places $P' \subseteq P$ whose successors are also predecessors: $P'^{\bullet} \subseteq {}^{\bullet}P'$.

A *siphon* is a non-empty set of places $P' \subseteq P$ whose predecessors are also successors: ${}^{\bullet}P' \subseteq P'^{\bullet}$.

A siphon (resp. a trap) is *proper* if its predecessor set is strictly included in its successor set: ${}^{\bullet}P' \subsetneq P'^{\bullet}$ (resp. $P'^{\bullet} \subsetneq {}^{\bullet}P'$).

A siphon (resp. a trap) is *minimal* if it does not contain any other siphon (resp. trap).

It is worth remarking that a siphon in N is a trap in the dual Petri net graph, obtained by reversing the direction of all arcs in N. Note also that since

predecessors and successors of an union are the union of predecessors (resp. successors), the union of two siphons (resp. traps) is a siphon (resp. a trap).

The Petri net of Example 1 has two minimal siphons, $\{A, AE\}$ and $\{E, AE\}$. $\{E, AE\}$ is both a minimal siphon and a minimal trap since ${}^\bullet\{E, AE\} = \{E, AE\}^\bullet = \{t_1, t_{-1}, t_2\}$

Although siphons and traps are stable under union, it is worth noting that minimal siphons do not form a generating set of all siphons. A siphon is called a *basis* siphon if it cannot be represented as a union of other siphons [35]. Obviously, a minimal siphon is also a basis siphon, however, not all basis siphons are minimal. For instance, in Example 1, the generating set of siphons is formed by $\{A, AE\}$, $\{E, AE\}$, $\{A, AE, B\}$ and $\{E, AE, B\}$, but only $\{A, AE\}$ and $\{E, AE\}$ are minimal, the two others cannot be obtained by union of minimal siphons.

The following propositions show that traps and siphons provide a structural characterization of some particular dynamical properties on markings.

Proposition 1 *[40] For every subset $P' \subseteq P$ of places, P' is a trap if and only if for any marking $m \in \mathbb{N}^P$ with $m_p \geq 1$ for some place $p \in P'$, and any marking $m' \in \mathbb{N}^P$ such that $m \xrightarrow{\sigma} m'$ for some sequence σ of transitions, there exists a place $p' \in P'$ such that $m'_{p'} \geq 1$.*

Proposition 2 *[40] For every subset $P' \subseteq P$ of places, P' is a siphon if and only if for any marking $m \in \mathbb{N}^P$ with $m_p = 0$ for all $p \in P'$, and any marking $m' \in \mathbb{N}^P$ such that $m \xrightarrow{\sigma} m'$ for some sequence σ of transitions, we have $m'_{p'} = 0$ for all $p' \in P'$.*

2.3 Application to Deadlock Detection

One reason to consider minimal siphons is that they provide a sufficient condition for the non-existence of deadlocks. A deadlock occurs in a marked Petri net if no transition is enabled. It has been shown indeed that in a deadlocked (and marked) Petri net, all unmarked places form a siphon [6]. The siphon-based approach for deadlocks detection checks if the net contains a proper siphon that can become unmarked by some firing sequence. In parrticular the following STP property provides a sufficient condition for ordinary Petri nets to be deadlock free.

Definition 11 Given a Petri net (P, T, W, m_0), the *siphon-trap property (STP)* holds when every siphon contains a marked trap.

Theorem 1 ([7]) *An ordinary Petri net in which the STP holds is deadlock free.*

Proof. We just have to show that STP is preserved by transition firing and that a dead marking does not satisfy STP. The preservation follows from the fact that a marked trap remains marked after the firing of a transition. If a marking m is dead, then the set $S = \{p \in P \mid m(p) = 0\}$ is such that $S^\bullet = T$

since every transition should have an empty predecessor. Then ${}^\bullet S \subseteq S^\bullet$, and moreover S is non-empty since T is non-empty. Therefore S is a siphon and does not contain a marked trap. □

The relevance of siphons and traps for other liveness properties in systems biology are reported in [23].

2.4 Application to Systems Biology

One example of the relevance of traps and siphons in biology was given in [49] for the analysis of the production and accumulation of starch in potato tubers during growth, while starch is consumed after the tubers are deposited after the harvest. This can be seen by a purely structural analysis of the Petri net graph of the metabolic network, since starch and several of its precursors form traps in the reaction net during growth, while starch and possible intermediates of degradation form siphons after the harvest. A simplified version of this Petri net is depicted in Figure 2, where G_1 stands for glucose-1-phosphate, G_u is UDP-glucose, S is the starch, I stands for intermediary species and P_1 and P_2 represent external metabolites [45]. In this network, either the branch producing starch (t_3 and t_4) or the branch consuming it (t_5 and t_6) is operative, as it is shown in Figure 3 and Figure 4 respectively. This is realized by a switching mechanism in the gene regulatory network with synthesis of the corresponding enzymes. Two Petri nets are thus derived from this model: one Petri net where t_5 and t_6 are removed (in this Petri net, t_3 and t_4 are said to be operative) and one Petri net where t_3 and t_4 are removed.

It can be easily observed that the set $\{G_u, S\}$ is a trap when t_3 and t_4 are operative: once a token arrives in S, no transition can be fired and the token remains there independently of the evolution of the system. Dually, $\{S, I\}$ is a siphon when t_5 and t_6 are operative: once the last token is consumed from S and I, no transition can generate a new token in these places, so they remain empty.

Another interesting example of use of the concept of siphons and traps, also from [49], deals with the analysis of the role of the triosephosphate isomerase (TPI) in Trypanosoma brucei metabolism. Earlier, Helfert et al. [24] supposed that glycolysis could proceed without TPI, but unexpected evidence that all system fluxes (Pyruvate, Glycerol) decrease was found which lead the authors to build a kinetic model for explaining that phenomenon. However, a purely structural explanation for the necessary presence of TPI in glycolysis and glycerol production was provided in [49] simply by showing the existence of siphons and traps in the model.

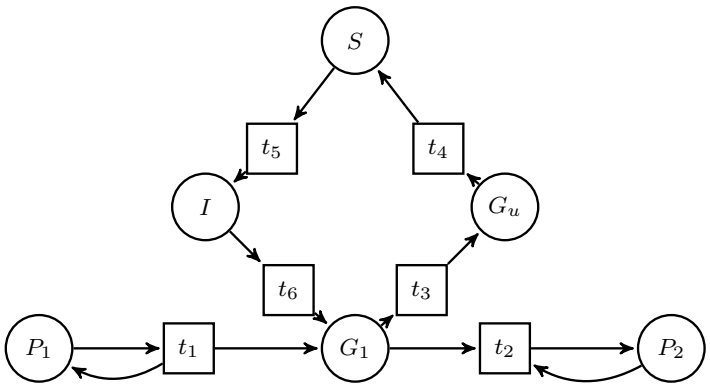


Fig. 2: Petri net graph modeling the growth metabolism of the potato plant [49].

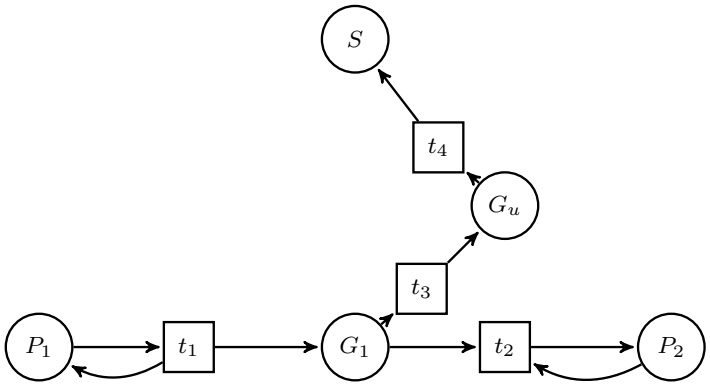


Fig. 3: Petri net graph modeling the growth metabolism of the potato plant [49] with the branch producing starch operative.

3 Theoretical Complexity of Siphon and Trap Properties

3.1 Preliminaries on SAT, FO and MSO

We shall prove some NP-completeness results using the following characterization of Boolean satisfiability. For a finite set of variables $V = \{x_1, \dots x_m\}$, let $\neg V = \{\neg x_1, \dots, \neg x_m\}$ denotes the set of negative literals constructed upon V. For a Boolean formula in conjunctive normal form $\phi = c_1 \wedge \cdots \wedge c_n$ over V, we have for all $1 \leq i \leq n$, $c_i = \ell_{i,1} \vee \cdots \vee \ell_{i,n_i}$, and for all $1 \leq j \leq n_i$, $\ell_{i,j} \in V \cup \neg V$. Let us write $C_\phi = \{i \in \mathbb{N} \mid 1 \leq i \leq n\}$ and $L_\phi = \{(i,j) \in \mathbb{N}^2 \mid 1 \leq i \leq n, 1 \leq j \leq n_i\}$.

Lemma 1 *A Boolean formula ϕ in conjunctive normal form is satisfiable if and only if there exists a subset $X \subseteq L_\phi$ such that*

- *for all $1 \leq i \leq n$, there exists $1 \leq j \leq n_i$ such that $(i,j) \in X$,*

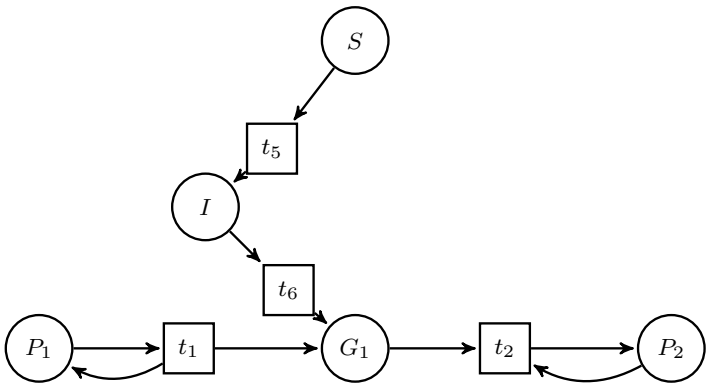


Fig. 4: Petri net graph modeling the growth metabolism of the potato plant [49] with the branch consuming starch operative.

– *for all* $(i,j),(i',j') \in X$, $\ell_{i',j'} \neq \neg\ell_{i,j}$.

Proof. If $\mu : V \to \{0,1\}$ satisfies ϕ, we pose for all $x \in V$, $\mu(\neg x) = 1 - \mu(x)$ and then it suffices to observe that $X = \{(i,j) \in L \mid \mu(\ell_{i,j}) = 1\}$ satisfies the conditions of the lemma. Conversely, given a subset $X \subseteq L_\phi$ satisfying these conditions, we pose $\mu : V \to \{0,1\}$ such that for all $x \in V$, $\mu(x) = 1$ if there exists $(i,j) \in X$ such that $\ell_{i,j} = x$ and 0 otherwise. Then, we observe that μ satisfies ϕ. □

We say that ϕ *is satisfied by* X if X is a subset of L_ϕ satisfying the conditions of Lemma 1.

We shall also use the language of first-order logic (FO) to express properties over finite set and graph structures assumed to be fixed. For each finite set S, we assume a unary predicate $S(x)$, also written $x \in S$ by abuse of notation, which tests membership. Non-emptyness, $S \neq \emptyset$, is expressed by the FO formula $\exists x(x \in S)$, and set inclusion, $S \subseteq S'$, by $\forall x(x \in S \to x \in S')$. Similarly for each Petri net graph, we assume two unary (set) predicates, place(x) and transition(x), which distinguish between places and transitions, and a binary predicate, edge(x,y), which tests incidence. This leads to the following characterization of siphons and traps:

Definition 12 The siphon and trap properties can be defined in FO by the following schemas of formulae:

$$\begin{aligned} \textsc{Siphon}(S): \; & S \neq \emptyset \wedge \forall p \in S \ \mathrm{place}(p) \\ & \wedge \forall t(\exists p \in S \ \mathrm{edge}(t,p) \to \exists p \in S \ \mathrm{edge}(p,t)) \\ \textsc{Trap}(S): \; & S \neq \emptyset \wedge \forall p \in S \ \mathrm{place}(p) \\ & \wedge \forall t(\exists p \in S \ \mathrm{edge}(p,t) \to \exists p \in S \ \mathrm{edge}(t,p)) \end{aligned}$$

We shall also provide a series of linear time complexity results by showing that some problems can be expressed in monadic second-order logic (MSO) over

finite graph structures, in order to use Courcelle's theorem. MSO extends FO by the addition of second-order quantifiers over predicates, also noted $\exists$ and $\forall$, with the restriction to apply to unary predicates only, i.e. sets. An example of MSO formula is given in the proof of Theorem 4 in the next section.

Definition 13 ([42]) A *tree decomposition* of a non-oriented graph $G = (V, E)$ is a pair (X, T) where $T = (I, A)$ is a tree and $X = (X_i)_{i \in I}$ is a family of subsets of V such that

- $\bigcup_{i \in I} X_i = V$,
- for all $\{v, v'\} \in E$, there exists $i \in I$ such that $\{v, v'\} \subseteq X_i$,
- for all $i, j, k \in I$, if j lies on the path from i to k in T, then $X_i \cap X_k \subseteq X_j$.

The *tree-width of a tree decomposition* is $\max_{i \in I} |X_i| - 1$. The *tree-width* $\mathrm{tw}(G)$ *of* G is the minimum tree-width taken over all possible tree decompositions of G.

The tree-width of an ordinary Petri net graph $N = (P, T, W)$ follows that definition given for graphs, by reading the Petri net as the non-oriented graph $G = (P \cup T, E)$ where $E = \{\{p, t\} \subseteq P \cup T \mid W(p, t) > 0 \text{ or } W(t, p) > 0\}$.

Example 2 Let us consider the Petri net graph depicted in Figure 5. The non-oriented graph associate to this Petri net is depicted in Figure 6. Two possible tree decompositions of this non-oriented graph are shown in Figure 7. In both decompositions, each graph edge connects two vertices which belong to the same tree node. Graph vertices are adjacent only when the corresponding subtrees intersect. In the first tree decomposition (Figure 7 on the left), each tree node contains at most four vertices, hence the width of this decomposition is three. In the second tree (Figure 7 on the right), each tree node contains at most three vertices, hence the width of this decomposition is two, which is the optimal tree-width over all possible tree decompositions. Hence, the tree-width of the Petri net graph depicted in Figure 5 equals two.

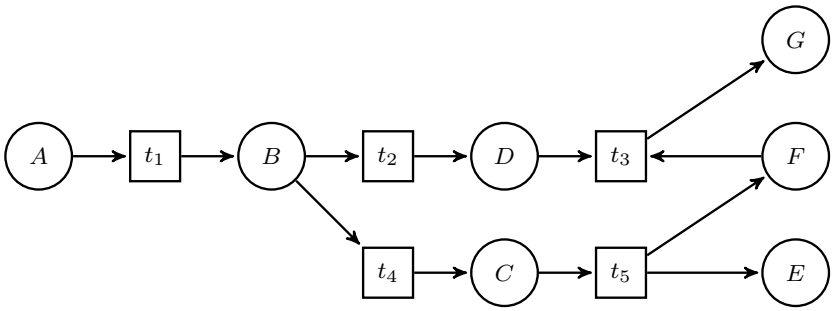


Fig. 5: Example of a Petri net with its associated non-oriented graph depicted in Figure 6 and two tree decompositions given in Figure 7.

Courcelle's theorem states that every graph property definable in MSO can be decided in linear time on graphs of bounded tree-width.

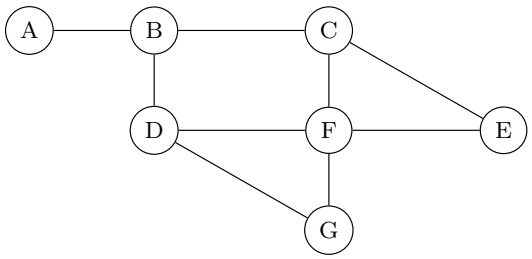


Fig. 6: Non-oriented graph associated to the Petri net of Figure5

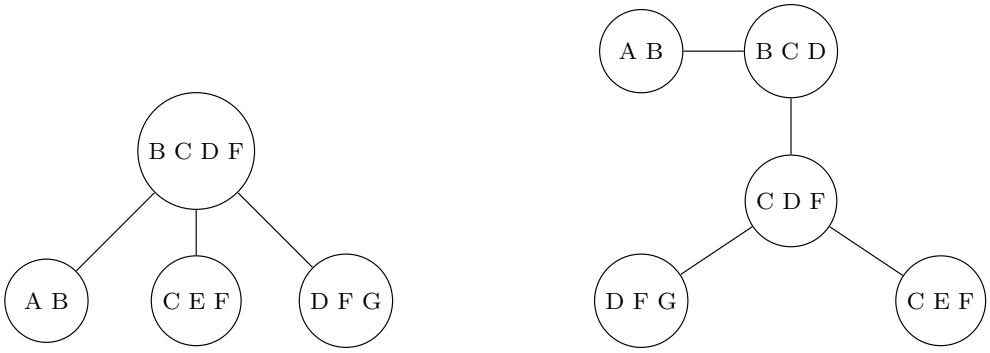


Fig. 7: Two possible tree decompositions of the graph shown in Figure 6

Theorem 2 (Courcelle [12]) *For a given formula P in monadic second-order logic on the structure of graphs and for a given positive integer k, there exists a linear time algorithm that given a finite graph G of tree-width at most k decides whether G satisfies P.*

Given a tree-automaton constructed from the formula, and a tree constructed from the graph decomposition that minimizes the tree-width, deciding if the automaton accepts the tree can be done in linear time. Courcelle's theorem shows the existence of such automata for MSO properties. However, the automata can be of hyper-exponential sizes. This makes the linear algorithm usually unusable in practice. There is some recent work in [13,28] on fly-automata and game-theoretic methods, aiming at making this result practical for certain properties of graphs with bounded clique-width. The applicability of these new approaches to our problem is however beyond the scope of this paper.

3.2 The Q-MinimalSiphon Problem

Definition 14 The problem FindMinimal is the following problem: "*given a Petri net graph $N = (P, T, W)$ and a subset of places $Q \subseteq P$, find a minimal siphon S in N such that $Q \subseteq S$ if there is any, or fail.*"

Theorem 3 ([46]) *The problem* FindMinimal *is polynomial.*

Definition 15 Q-MinimalSiphon is the following decision problem: "*given a Petri net graph $N = (P, T, W)$ and a subset of places $Q \subseteq P$, does there exist a minimal siphon S in N such that $Q \subseteq S$?*".

Theorem 4 Q-MinimalSiphon *is decidable in linear time for any class of Petri net graphs with bounded tree-width.*

Proof. By Definition 12, siphons and traps are expressible in FO, and thus in MSO. We just have to verify that Q-MinimalSiphon can be expressed in MSO as follows:

$$\begin{aligned} Q\text{-MinimalSiphon} : \exists S(&\textsc{Siphon}(S) \wedge Q \subseteq S \\ &\wedge \forall S'(\textsc{Siphon}(S') \wedge S' \subseteq S \rightarrow S \subseteq S')) \end{aligned}$$

The linear time complexity then follows from Courcelle's theorem. □

In the general case, Q-MinimalSiphon has been shown NP-complete in [48]. We find it useful here to provide a simpler proof of this result, by showing the NP-hardness of the following equivalent problem.

Definition 16 For a given Petri net graph $N = (P, T, W)$ and $Q \subseteq P$, a *Q-hitting siphon* is a siphon $S \subseteq P$ of N such that for every siphon $S' \subseteq S$ of N, $Q \subseteq S'$.

Definition 17 Q-HittingSiphon is the following decision problem: "*given a Petri net graph $N = (P, T, W)$ and a subset of places $Q \subseteq P$, does there exist a Q-hitting siphon in N?*".

Proposition 3 *Given a Petri net graph $N = (P, T, W)$ and a subset of places $Q \subseteq P$, there exists a minimal siphon containing Q in N if and only if there exists a Q-hitting siphon in N.*
Proof. If S is a minimal siphon containing Q, then S is a Q-hitting siphon. Conversely, if S is a Q-hitting siphon, then there exists a minimal siphon $S' \subseteq S$ and since S is Q-hitting, then $Q \subseteq S'$. □

Theorem 5 Q-MinimalSiphon *is NP-complete.*
Proof. Q-MinimalSiphon is in NP since FindMinimal is polynomial. We just have to show that Q-HittingSiphon is NP-hard.

Let us assume a SAT instance described by a set V of variables and a Boolean formula ϕ over V in conjunctive normal form. Let $N = (P, T, W)$ be the ordinary Petri net graph, depicted in Figure 8, where

- $P = L_\phi \cup \{q\}$ and $T = V \cup \neg V \cup C_\phi$ where $q \notin L_\phi \cup T$,
- W satisfies

$$\begin{array}{cc} {}^\bullet(i,j) = \{\neg \ell_{i,j}\} & (i,j)^\bullet = \{i, \ell_{i,j}\} \quad \text{for all } (i,j) \in L_\phi \\ {}^\bullet q = C_\phi & q^\bullet = V \cup \neg V \end{array}$$

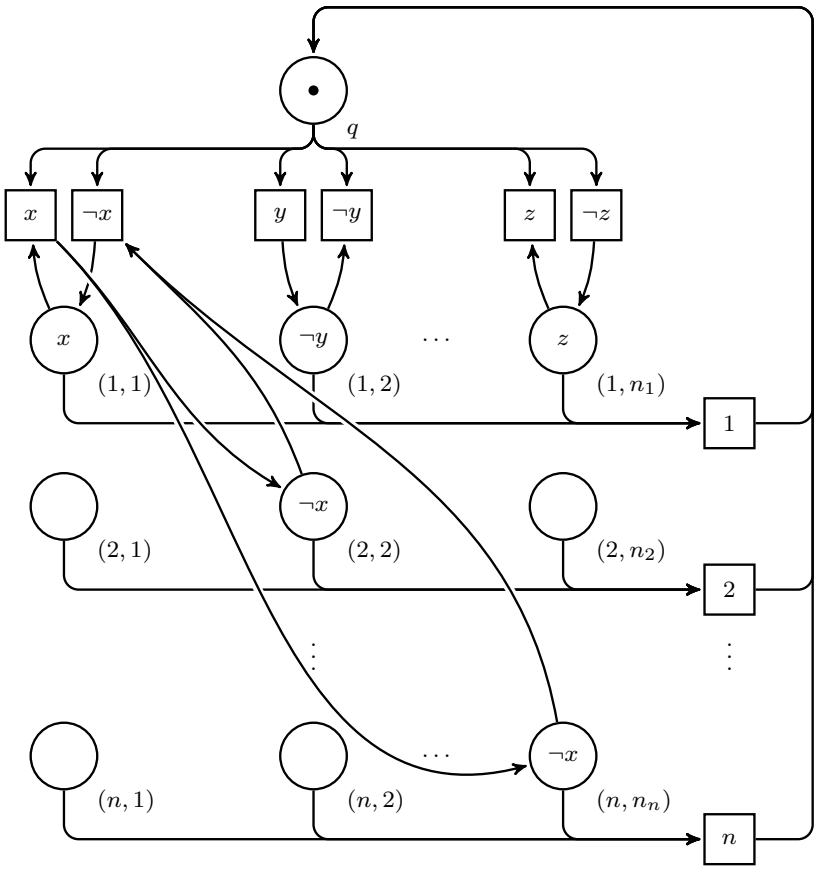


Fig. 8: Petri net used in the reduction of SAT to Q-HittingSiphon in Theorem 5. In this Petri net, a siphon S contains the place q on top of the figure if and only if S contains at least one literal occurrence per clause. Moreover, S is $\{q\}$-hitting if and only if S does not contain a pair of contradictory literals.

Lemma 2 *ϕ is satisfiable if and only if there exists a $\{q\}$-hitting siphon in N, i.e., a subset $S \subseteq P$ such that $S \neq \emptyset$ and ${}^\bullet S \subseteq S^\bullet$ and for all $S' \subseteq S$, if $S' \neq \emptyset$ and ${}^\bullet S' \subseteq S'^\bullet$, then $q \in S'$.*

Proof. Suppose that ϕ is satisfied by $X \subseteq L_\phi$. We verify that $X \cup \{q\}$ is a $\{q\}$-hitting siphon. Indeed, ${}^\bullet X \subseteq V \cup \neg V = q^\bullet$ and, by hypotheses on X, ${}^\bullet q = C_\phi \subseteq X^\bullet$, therefore ${}^\bullet(X \cup \{q\}) \subseteq (X \cup \{q\})^\bullet$. For any $S' \subseteq S$ such that $S' \neq \emptyset$ and ${}^\bullet S' \subseteq S'^\bullet$, suppose that $q \notin S'$, then since $S' \neq \emptyset$, there exists $(i,j) \in S'$ and ${}^\bullet(i,j) = \{\neg\ell_{i,j}\} \subseteq S'^\bullet$, therefore there exists $(i',j') \in S'$ such that $\neg\ell_{i,j} \in (i',j')^\bullet$, therefore $\ell_{i',j'} = \neg\ell_{i,j}$, which contradicts that $S' \subseteq X$; thus, $q \in S'$. That proves that $X \cup \{q\}$ is a $\{q\}$-hitting siphon.

Conversely, suppose that S is a $\{q\}$-hitting siphon. We verify that ϕ is satisfied by $X = S \cap L_\phi$. Indeed, for all $1 \leq i \leq n$, $i \in {}^\bullet q$, then $i \in X^\bullet$, therefore there exists $1 \leq j \leq n_i$ such that $(i,j) \in X$. Suppose that there exist $(i,j),(i',j') \in X$ such that $\ell_{i',j'} = \neg\ell_{i,j}$, then $S' = \{(i,j),(i',j')\}$ would be

such that $S' \neq \emptyset$ and ${}^\bullet S' \subseteq S'^\bullet$, despite $q \notin S'$. Therefore, for all $(i,j),(i',j') \in X$, $\ell_{i',j'} \neq \neg\ell_{i,j}$. Thus, ϕ is satisfiable. □

The theorem is then an immediate consequence of the lemma. □

3.3 The k-SIPHON Problem

Definition 18 The problem k-SIPHON is the following decision problem: "*given a Petri net graph $N = (P,T,W)$ and a positive integer k, does there exist a siphon S in N of cardinality k?*".

The linear time complexity immediately follows from Courcelle's theorem.

Theorem 6 *k-SIPHON is decidable in linear time with respect to the size of the Petri net for any class of Petri net graphs with a bounded tree-width.*
Proof. Given a Petri net graph $N = (P,T,W)$ and a positive integer k, there exists a siphon S in N of cardinality k if and only if $\exists S(\textsc{Siphon}(S) \wedge \mathrm{card}_k(S))$ is satisfied, where the formula $\mathrm{card}_k(S)$

$$\mathrm{card}_k(S) : \exists x_1 \dots x_k \left(\bigwedge_{1 \leq i < j \leq k} x_i \neq x_j \wedge \forall x \left(x \in S \rightarrow \bigvee_{1 \leq i \leq k} x = x_i \right) \right)$$

checks that the cardinality of S is k. □

We prove the NP-completeness of k-SIPHON by polynomial reduction from the set covering problem, one of Karp's original NP-complete problems [26]. Let us recall that the problem k-SETCOVERING is the following decision problem: "*given a finite set $\mathcal{U}$ (the universe), a subset $\mathcal{S}$ of $\mathfrak{P}(\mathcal{U})$ and an integer k, does there exist a subset $S \subseteq \mathcal{S}$ of cardinality k such that $\mathcal{U} = \bigcup S$?*".

Theorem 7 *k-SIPHON is NP-complete.*
Proof. k-SIPHON is in NP since checking that a given set of places is a siphon of cardinality k is polynomial. NP-hardness comes by polynomial reduction from k-SETCOVERING: given a finite set $\mathcal{U}$ (the universe), a subset $\mathcal{S}$ of $\mathfrak{P}(\mathcal{U})$ and an integer k, let $N = (P,T,W)$ be the ordinary Petri net graph such that $P = \mathcal{S}$, $T = \mathcal{U}$ and for all $t \in T$, $t^\bullet = P$ and ${}^\bullet t = \{S \in \mathcal{S} \mid t \in S\}$. Then for every subset $S \subseteq P$, S is a siphon if and only if $\mathcal{U} = \bigcup S$. □

This shows that the optimization problem MINIMALCARDINALITYSIPHON is NP-hard.

3.4 The STP Problem

Definition 19 The Siphon-Trap Property STP is the following decision problem: "*given a marked Petri net $N = (P, T, W, m_0)$, does every siphon in N include a trap that contains a marked place?*".

Theorem 8 STP *is decidable in linear time for any class of Petri nets with bounded tree-width.*

Proof. Here again, it suffices to remark that STP is expressible in MSO, with a unary predicate marked(p) for distinguishing marked places, and the following MSO formula:

$$\text{STP} : \forall S(\textsc{Siphon}(S) \rightarrow \exists T(\textsc{Trap}(T) \wedge T \subseteq S \wedge \exists p(\text{marked}(p) \wedge p \in T)))$$

The linear time complexity immediately follows from Courcelle's theorem. □

In the general case, STP has been shown to be in co-NP in [39], by reducing ¬STP to SAT. Indeed, ¬STP expresses the existence of a siphon S such that every trap included in S does not intersect M. The encoding in SAT focuses on the maximal trap included in S (the union of all the traps included in S), which is computed by removing iteratively places in S that cannot belong to a trap.

In fact, we can prove

Theorem 9 STP *is co-NP-complete.*

Proof. Since STP is in co-NP [39], it suffices to show that ¬STP is NP-hard.

Let us assume a SAT instance described with a set V of variables and a Boolean formula ϕ over V in conjunctive normal form. Let $N = (P, T, W, m_0)$ be the ordinary Petri net, depicted in Figure 9, where

- $P = L_\phi \cup \{0\} \times V$ and $T = V \cup \neg V \cup C_\phi \cup \{t\}$ where $t \notin P \cup V \cup \neg V \cup C_\phi$,
- W satisfies

$$\begin{array}{lll} {}^\bullet(i,j) = C_\phi \cup \{\ell_{i,j}\} & (i,j)^\bullet = \{i,t\} & \text{for all } (i,j) \in L_\phi \\ {}^\bullet(0,x) = \{t\} & (0,x)^\bullet = \{x, \neg x\} & \text{for all } x \in V \end{array}$$

- $m_0 = \mathbb{1}_{\{0\} \times V}$.

Note that the set $\{0\} \times V$ is introduced in places to ensure that $P \cap T = \emptyset$.

Lemma 3 *ϕ is satisfiable if and only if $(N, \{0\} \times V)$ satisfies* ¬STP*,* i.e.*, there exists a subset $S \subseteq P$ such that $S \neq \emptyset$ and ${}^\bullet S \subseteq S^\bullet$ and for all $T \subseteq S$, if $T \neq \emptyset$ and $T^\bullet \subseteq {}^\bullet T$, then $T \cap \{0\} \times V = \emptyset$.*

Proof. Suppose that ϕ is satisfied by a set $X \subseteq L_\phi$. We verify that $X \cup \{0\} \times V$ is a siphon and that it does not contain any trap intersecting $\{0\} \times V$. Indeed, by hypotheses on X, $C_\phi \subseteq X^\bullet$, and ${}^\bullet(X \cup \{0\} \times V) \subseteq C_\phi \cup V \cup \neg V \cup \{t\} \subseteq (X \cup \{0\} \times V)^\bullet$. For any $T \subseteq X \cup \{0\} \times V$ such that $T \neq \emptyset$ and $T^\bullet \subseteq {}^\bullet T$, suppose that there exists $x \in T \cap \{0\} \times V$, then $\{x, \neg x\} \subseteq T^\bullet$, therefore $\{x, \neg x\} \subseteq {}^\bullet T$ and there exist $(i,j), (i',j') \in T \cap X$ such that $\ell_{i',j'} = \neg \ell_{i,j}$,

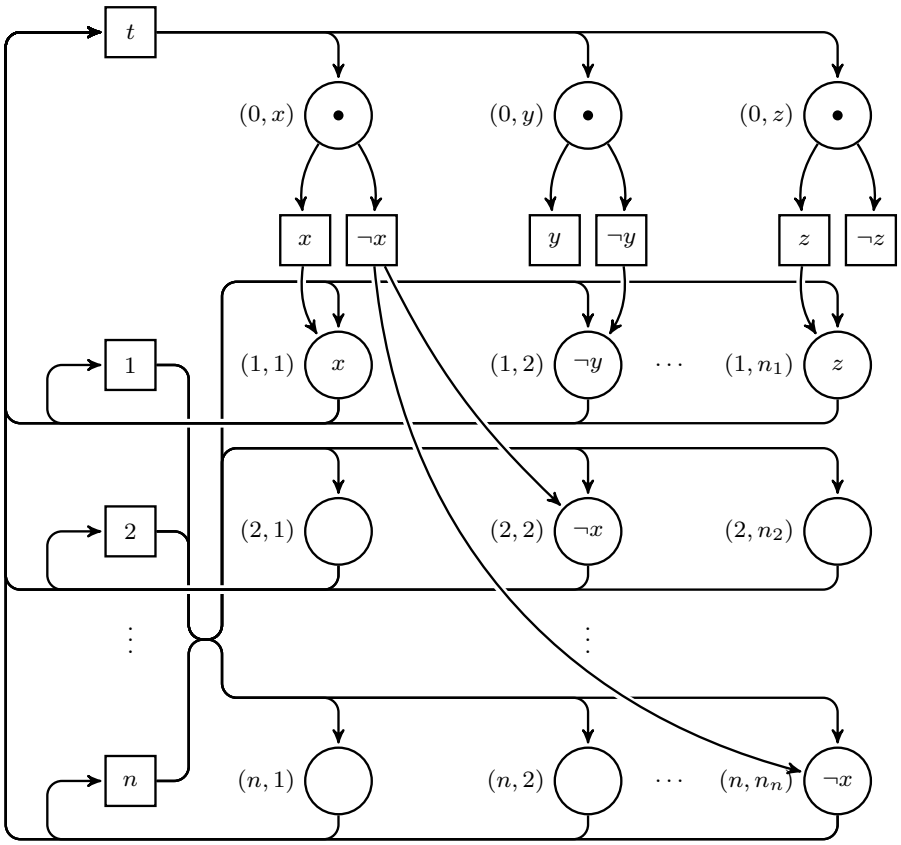


Fig. 9: Petri net used in Theorem 9 for the reduction of SAT to STP. In this Petri net set of places S is a siphon if and only if (1) S contains at least one literal occurrence per clause and (2) S contains all the places $(0, x)$ such that the variable x occurs in these literal occurrences; moreover, a marked place $(0, x)$ belongs to a trap included in S if and only if S contains the pair of contradictory literal occurrences x and $\neg x$.

that would contradict the hypotheses on X. Therefore $(N, \{0\} \times V)$ satisfies $\neg$STP.

Conversely, suppose that there exists a subset $S \subseteq P$ such that $S \neq \emptyset$ and ${}^\bullet S \subseteq S^\bullet$ and for all $T \subseteq S$, if $T \neq \emptyset$ and $T^\bullet \subseteq {}^\bullet T$, then $T \cap \{0\} \times V = \emptyset$. We verify that ϕ is satisfied by $X = S \cap L_\phi$. We have $X \neq \emptyset$ because otherwise there would exist $(0, x) \in S \cap \{0\} \times V$, then $t \in {}^\bullet S$, therefore $t \in S^\bullet$ and there would exist $(i, j) \in S \cap L_\phi$, which contradicts $X = \emptyset$. Thus, $C_\phi \subseteq {}^\bullet S \subseteq S^\bullet$ and for all $1 \leq i \leq n$, $i \in S^\bullet$ so there exists $1 \leq j \leq n_i$ such that $(i, j) \in X$. Suppose that there exist $(i, j), (i', j') \in X$ such that $\ell_{i',j'} = \neg \ell_{i,j}$. Then, there exists $(0, x) \in S \cap \{0\} \times V$ such that $(0, x)^\bullet = \{\ell_{i,j}, \ell_{i',j'}\}$. Then $T = \{(0, x), (i, j), (i', j')\}$ will be such that $T^\bullet = \{\ell_{i,j}, \ell_{i',j'}, i, i', t\} \subseteq C_\phi \cup \{\ell_{i,j}, \ell_{i',j'}, t\} = {}^\bullet T$, but $T \cap \{0\} \times V \neq \emptyset$. Therefore, for all $(i, j), (i', j') \in X$, $\ell_{i',j'} \neq \neg \ell_{i,j}$. □

The theorem is now an immediate consequence of the lemma. □

4 Boolean Model for Minimal Siphons

The definition of siphons in FO given in Definition 12 directly leads to the following Boolean Constraint Satisfaction Problem (CSP):

Definition 20 Given a Petri net graph $N = (P, T, W)$, the CSP SIPHON(N) is the triple (V, D, C) where

- $V = P$, i.e. a variable is introduced for each place,
- $D(p) = \{0, 1\}$ for all $p \in V$, i.e. the variables are Boolean,
- $\bigvee_{p \in S} p = 1$, i.e. a siphon is not empty,
- $C = \{(p = 1 \rightarrow \bigvee_{p' \in {}^\bullet t} p' = 1) \mid p \in P, t \in {}^\bullet p\}$.

Proposition 4 *For every Petri net graph $N = (P, T, W)$, the CSP* SIPHON(N) *is satisfied by a valuation ν if and only if $\{p \in P \mid \nu(p) = 1\}$ is a siphon.*
Proof. It suffices to check that for every non-empty set of places S, we have $\forall p, \forall t \in {}^\bullet p, p \in S \rightarrow \bigvee_{p' \in {}^\bullet t} p' \in S$ if and only if ${}^\bullet S \subseteq S^\bullet$. □

The encoding in a SAT solver is short and direct. For each transition t in the set of predecessors of a place p, a clause C is added to the satisfiability problem. C is formed by a negated boolean variable associated to p or-ed with boolean variables in the set of predecessors of t. To avoid the trivial case of the empty siphon, one clause is added.

The set of all minimal siphons (*w.r.t.* set inclusion) can be enumerated in the set inclusion order, by restarting search each time a siphon S is found, with the additional constraint $\bigvee_{p \in S} p = 0$, for disallowing any superset of that siphon.

For enumerating siphons in set inclusion order, we compared two techniques: one by labeling an auxiliary cardinality variable in increasing order ([36]), one by labeling directly the Boolean variables with increasing value selection (to test first the absence, then the presence of a place in the candidate solution). The second technique has revealed to be much more efficient. The correctness of this technique comes from the fact that if a solution S' is found after a solution S, then the two paths in the search tree leading to these solutions have a least common ancestor node: this node corresponds to the labeling of a place p. By construction, p belongs to S' and not to S, thus $S' \not\subseteq S$.

We shall also consider a variant of the above CSP where the constraints $p = 1 \rightarrow \bigvee_{p' \in {}^\bullet t} p' = 1$ are decomposed by introducing an intermediary Boolean variable for each transition. For every Petri net graph $N = (P, T, W)$, the CSP SIPHON'(N) $= (V, D, C)$ is defined as follows.

- $V = P \cup T$, i.e. one variable is introduced for each place and each transition,

- $D(x) = \{0, 1\}$ for all $x \in V$,
- $\bigvee_{p \in S} p = 1$, i.e. a siphon is not empty,
- $C = \{(p = 1 \to t = 1) \mid p \in P, t \in {}^\bullet p\} \cup \{(t = 0 \to p = 0) \mid t \in T, p \in {}^\bullet t\}$.

It is immediate that if SIPHON'(N) is satisfied by a valuation ν, then SIPHON(N) is satisfied by $\nu|_P$, and conversely that if SIPHON(N) is satisfied by ν, then ν can be extended to a valuation satisfying SIPHON'(N).

This variant SIPHON'(N) enjoys the following property.

Proposition 5 *For every Petri net graph $N = (P, T, W)$, the CSP* SIPHON'(N) *has the same tree-width as N.*

Proof. It suffices to notice that the primal graph of SIPHON'(N) is isomorphic to the bipartite graph induced by N. □

5 Implementations with SAT and CLP Solvers

This section describes two implementations of the above model and search strategy, one using an iterated SAT procedure and the other based on Constraint Logic Programming with Boolean constraints.

5.1 Iterative SAT Algorithm

The Boolean model can be directly interpreted using a SAT solver to check the existence of a siphon or trap. We use sat4j [3], an efficient library of SAT solvers in Java for Boolean satisfaction and optimization. It includes an implementation of the MiniSAT algorithm, that implements the value selection mentioned above: for each variable, the value 0 is tried before the value 1.

Example 3 We consider the Petri net depicting the enzymatic reaction of Example 1. In the first iteration, the problem amounts to solve the following encoding of Horn-dual clauses: $(\neg A \vee AE) \wedge (\neg AE \vee E \vee A) \wedge (\neg E \vee AE) \wedge (\neg E \vee AE) \wedge (\neg B \vee AE)$ The problem is satisfied with the values: $E = B = 0$ and $A = AE = 1$, which means that $\{A, AE\}$ is a minimal siphon. In the second iteration, the clause $\neg A \vee \neg AE$ is added to ensure minimality, and the problem is satisfied with $A = B = 0$ and $E = AE = 1$ meaning that $\{E, AE\}$ is also a minimal siphon. A new clause is added stating that either E or AE does not belong to the siphon and no more variable assignment can satisfy the problem.

Therefore, this model contains two minimal siphons: $\{A, AE\}$ and $\{E, AE\}$. The enzyme E is a catalyst protein for the transformation of the substrate E in a product B. Such a catalyst increases the rate of the reaction but is conserved in the reaction.

[3] http://www.sat4j.org/

5.2 Backtrack Replay CLP(B) Algorithm

The search for siphons can also be implemented with a Constraint Logic Program with Boolean constraints (CLP(B)). We use GNU-Prolog [4] [17] for its efficient low-level implementation of Boolean constraint propagators.

The enumeration strategy is a variation of *branch-and-bound*, where the search is restarted to find a non-superset siphon each time a new siphon is found. We tried two variants of the branch-and-bound: with restart from scratch and by backtracking.

In the branch-and-bound with restart method, it is essential to choose a variable selection strategy which ensures diversity. Indeed, an enumeration method with a fixed variable order accumulates failures by always trying to enumerate the same sets first and these failures are only lately pruned by the non-superset constraints. As a consequence, the developed search tree gets more and more dense after each iteration since the previous forbidden sets are repeatedly tried again. This phenomenon does not exist in SAT solvers thanks to no-good recording. In CLP, random variable order selection strategy can be a good choice: this provides a good diversity and performs much better than any uniform heuristics.

However, branch-and-bound by backtracking gives better performance when care is taken for posting the non-superset constraint only once, since reposting it at each backtrack step proved to be inefficient. We have implemented a backtrack replay strategy, i.e. a customized branch and bound procedure where the search is performed as follows:

1. each time a siphon is found, the path leading to this solution is memorized,
2. then the search is fully backtracked to the root in order to add to the model the new non-superset constraint,
3. and then the memorized path is rolled back and replayed to continue the search at the point it was stopped.

Figure 10 (generated with CLPGUI [5] [19]) depicts the search tree that is developed for enumerating the 64 minimal siphons of a biological model of 51 species and 72 reactions. Each sub-tree immediately connected to the root corresponds to the replay of the path with a minimality constraint added. The small number of backtrack points shows the remarkable efficiency of the backtrack replay strategy combined with a simple Boolean constraint propagator.

6 Evaluation

In the literature, many algorithms have been proposed to compute minimal siphons and traps of Petri nets. Since a siphon in a Petri net N is a trap of the dual net N', it is enough to focus on siphons, the traps are obtained by duality. Some algorithms are based on linear programming [35, 9], Horn clause

[4] http://www.gprolog.org/

[5] http://contraintes.inria.fr/∼fages/CLPGUI

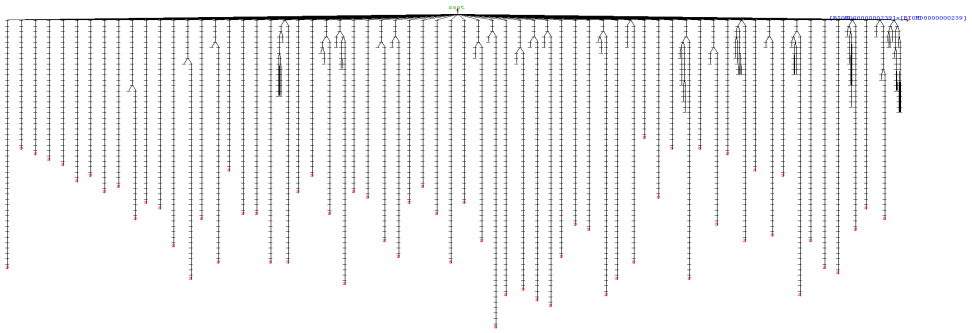


Fig. 10: Search tree developed with the backtrack replay strategy for enumerating the 64 minimal siphons of the model 239 of BioModels which contains 51 species and 72 reactions. The branches ending with a cross are solutions. It is remarkable that very few backtracks are necessary to enumerate all solutions using the backtrack replay strategy.

satisfaction [27,33] or algebraic approaches [31]. More recent state-of-the-art methods are presented in [10,11] and show the better performance of the dedicated algorithm of [11]. This algorithm uses a recursive problem partitioning procedure to reduce the original search problem to multiple simpler search sub-problems. Each sub-problem has specific additional place constraints with respect to the original problem. This algorithm can be applied to enumerate minimal siphons, place-minimal siphons, or even siphons that are minimal with respect to a given subset of places. In this section we compare our results to that dedicated algorithm.

6.1 BioModels Benchmark

The BioModels [32] database is a repository of peer-reviewed, published, computational models of biological processes. These models are written in the Systems Biology Markup Language (SBML) exchange format and are mainly composed of reaction rules, with or without kinetics, and events. This resource allows biologists to store, search and retrieve models referenced in publications.

We consider the curated part of the repository BioModels version February 2013. Among these 404 curated models, 59 models do not hold any reaction (but events only), we thus consider the 345 reaction models from which a Petri net graph can be extracted from the structure of the reactions.

In addition, we consider the following complex biochemical models:

- a model of E2F/Rb signaling from [4] which contains 408 molecular species and 534 reactions, and will be shown to contain 74 minimal siphons,
- Kohn's map of the mammalian cell cycle control [29,5], which contains 509 species and 775 reactions, and will be shown to contain 80 minimal siphons.

6.2 Petriweb Benchmark

The database Petriweb [6] [22] is an attempt to make available a significant body of Petri nets examples in a public repository. They are written in the Petri Net Markup Language (PNML), an emerging standard format supported by many tools PetriWeb supports a restricted form of PNML, including flat, uncoloured nets, plus limited support for hierarchy The repository can be browsed with a web browser, and individual nets can be retrieved and uploaded in PNML.

The repository contains 80 Petri nets given with some interesting properties. The properties are defined by the repository administrator. The properties are intended to be checked automatically by external analysis tools.

6.3 Computational Results and Comparisons

In this section we compare the SAT method, using the MiniSAT solver implementation included in the SAT4J library, the CLP method, using GNU-Prolog, both described in Section 5, and the state-of-the-art dedicated algorithm of [11].

Table 1 presents the CPU times in milliseconds for enumerating the sets of all minimal siphons of the Petri nets in our benchmark in Petriweb and BioModels (except Model 175 as explained below). The CPU times have been obtained on a PC with an intel Core processor 2.20 GHz and 8 GB of memory. For each benchmark, we provide the total number of models, the minimal, maximal and average numbers of siphons, and the total computation time in milliseconds for enumerating all of them.

Benchmark	# model	# siphons min-max (avg.)	siphons size min-max (avg.)	total time (ms)		
				dedicated algorithm	MiniSAT	GNU Prolog
BioModels	345	0-64 (4.21)	1-413 (3.10)	19734	611	195
Petriweb	80	0-11 (2.85)	0-7 (2.03)	2325	156	6

Table 1: Computation time in milliseconds on the BioModels and Petriweb benchmarks.

Quite surprisingly, on all these practical instances, both MiniSAT and GNU-Prolog solve the minimal siphon enumeration problem, in less than one millisecond in average, with a slightly better average performance for the CLP(B) program over the SAT solver. Furthermore, MiniSAT and GNU-Prolog outperform the dedicated algorithm by one or two orders of magnitude.

However, one particular model, number 175 in BioModels, has very high computation time and was excluded from Table 1. Table 2 presents the performance figures obtained on this model and on three other hardest instances for

[6] http://www.petriweb.org/

which we also provide the number of places and transitions. Even if the model is quite large, e.g. for Kohn's map of the cell cycle control with 509 species and 775 reactions, the computation time for enumerating its 81 minimal siphons is astonishingly short: one millisecond only. On these hard instances, the SAT solver is faster than the CLP(B) program by one or two orders of magnitude, and is the only algorithm to solve the problem for model 175, in 137 seconds.

model	# siphons	# places	# transitions	dedicated algorithm	MiniSAT	GNU Prolog
Kohn's map	81	509	775	28	1	221
BIOMD000000175	3042	118	194	∞	137000	∞
BIOMD000000205	32	194	313	21	1	34
BIOMD000000239	64	51	72	2980	1	22

Table 2: Computation time in milliseconds on the hardest instances of biochemical networks.

Model 175 represents a quantitative model that relates the EGF and HRG stimulations of the ErbB receptors to the activation of ERK and AKt in MCF-7 breast cancer cells [2]. This is the first model to take into account both the ERK and AKt pathways, four ErbB receptors, and their simultaneous activation by two ligands. Previous models of ErbB (e.g. the model developed in [43]) were limited to a single ErbB receptor because of combinatorial complexity. As a result, the ErbB signaling network is highly connected and indeed the underlying Petri net contains the highest number of arcs, and of organizations as remarked in [18, 25], of the BioModels repository.

6.4 Clause Density Analysis

The enumeration of the set of all minimal siphons is a problem of enumeration of all the solutions of an NP-complete problem, so the question is: *why are the CSP-based algorithms for enumerating siphons so efficient on the existing benchmarks, even on large graphs from systems biology*?

One possible explanation could be obtained by considering the well-known phase transition phenomenon in 3-SAT. The probability that a random 3-SAT problem is satisfiable has been shown to undergo a sharp phase transition as the ratio of the number of clauses over the number of variables crosses the critical value of about 4.26 [34, 14], going from satisfiability to unsatisfiability with probability one, when the number of variables grows to infinity. It is in this region of the density that the SAT instances are difficult to decide, while before and after that density the instances are usually easy.

This density of SAT instances associated to the enumeration of all minimal siphons, grows during enumeration since clauses are added for each solution found. We can thus check whether the initial density of the 3-SAT instances

associated to BioModels instances are greater than the critical value of 4.26. If the initial density is above the critical value, it will remain above, and the instances are thus easy because there will be a small number of solutions. On the other hand, if the initial density is below the threshold value of 4.26, the computation time may be long because the threshold value were the 3-SAT hardest instances are may be traversed, and the clauses may be satisfiable with an exponential number of solutions.

The density of a SAT instance is:

$$\text{density} = \frac{\#clauses}{\#variables}$$

Considering our problem of enumerating minimal siphons of a general Petri net $PN = (P, T, W)$ on our $\mid \mathrm{P} \mid$ Boolean variable, initially we have $\sum_{t \in T} \mid t^{\bullet} \mid$ clauses plus one clause of non-empty siphon:

$$\text{density} = \frac{\sum_{t \in T} \mid t^{\bullet} \mid +1}{\mid \mathrm{P} \mid}$$

To transform a general SAT instance to a 3-SAT instance, we add μ clauses and μ variables:

$$\text{density}_{\text{3-SAT}} = \frac{\sum_{t \in T} \mid t^{\bullet} \mid +1 + \mu}{\mid \mathrm{P} \mid +\mu}$$

where

$$\mu = \sum_{t \in T} \max(0, \mid {}^{\bullet}t \mid -2)$$

The initial density distribution of all BioModels instances are illustrated in the histogram of Figure 11.

This histogram shows that the initial density is in fact below the critical value for the majority of models. The initial 3-SAT density of our hardest model number 175 equals 2.39. Since the density grows during enumeration by adding the clauses for minimality, the possibility to traverse the critical region of density exists. Density considerations thus do not suffice to explain why we are so efficient in enumerating all the solutions of an NP-complete problem in large classes of practical Petri nets.

6.5 Tree-Width Analysis

There is also a rich literature about the polynomial-time complexity of CSPs when the constraint hypergraph is bounded relatively to a variety of graph measures, including cutwidth and tree-width [21]. Since we have shown that both Q-MinimalSiphon and k-Siphon are expressible as CSPs, the existence of a polynomial algorithm for deciding these properties for Petri net graphs of

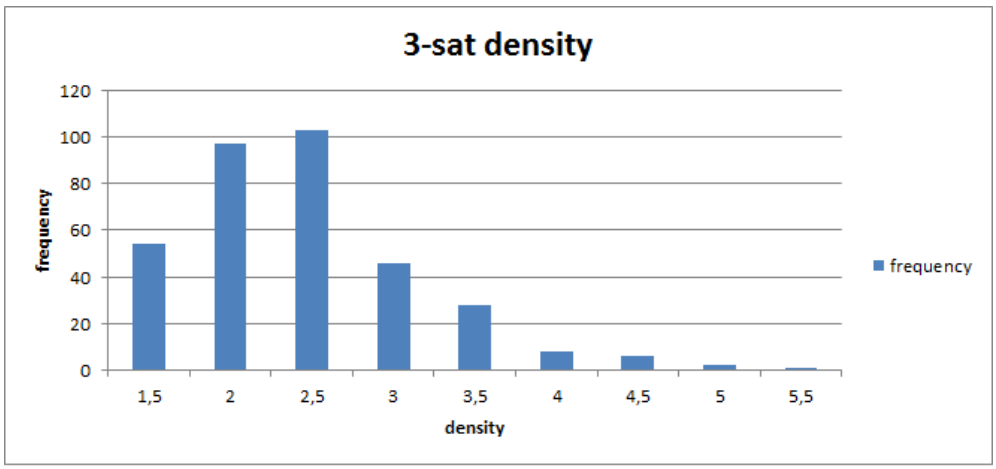


Fig. 11: Distribution of the initial density of the initial 3-SAT problems considered to enumerate all minimal siphons of the Petri net from the BioModels benchmark.

bounded tree-width follows by Prop. 5. However, since these problems are also expressible in Monadic Second Order (MSO) logic, they are in fact of linear time complexity in this case, as shown by Theorems 4 and 6.

It is thus interesting to measure the tree-widths of the Petri nets of our benchmark. QuickBB [7] is a program for computing the tree-width of a graph. When given enough time, this algorithm yields the exact tree-width of the graph. When stopped before termination, it yields an upper bound of the tree-width.

We have applied QuickBB on the 432 curated models of our BioModels benchmark. For 31 models, the exact tree-width could not be computed in a time-out of one hour, but the tree-width was bounded by 23. For the remaining 342 models, the exact tree-width was computed and was always less than 10 as shown in Figure 12. The tree-width of our three hardest instances are given in Table 3, and have been determined to be less than 10 and 15. These tree-width values are relatively small values for graphs of hundreds of species and reactions.

model	# places	# transitions	tree-width
BIOMD000000175	118	194	≤ 15
BIOMD000000205	194	313	≤ 10
BIOMD000000239	51	72	≤ 10

Table 3: Tree-width of the hardest instances of BioModels database.

[7] http://graphmod.ics.uci.edu/group/quickbb/

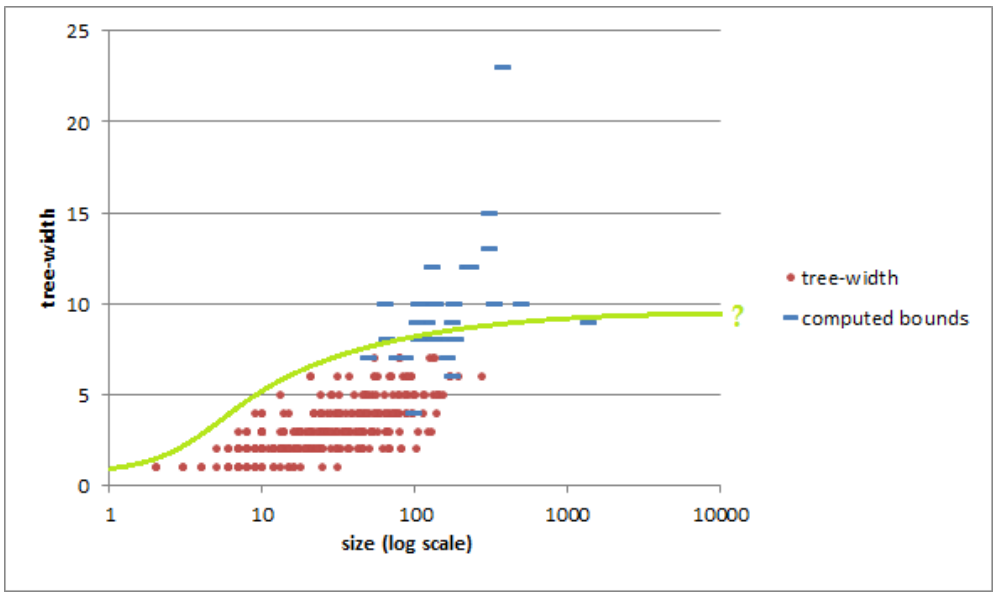


Fig. 12: Variation of the computed tree-widths as a function of size (places and transitions) of the Petri nets associated to BioModels. The computed exact tree-widths are depicted by red circles. When not known, the computed upper bounds are depicted by blue bars.

The BioModels benchmark thus seems to indicate that biochemical reaction networks have a bounded tree-width less than ten. This is in agreement with the idea that even very large biochemical processes are not fully interconnected as in a grid, but composed through interface molecular species. These considerations suggest that biochemical networks are of bounded tree-width, in which case Theorems 4, 6 and 8 show that the minimal siphon decision problems are indeed tractable, and in fact of linear time theoretical complexity.

7 Conclusion

Siphons and traps in Petri nets are meaningful pools of places that display a specific behaviour in the Petri net dynamics, and that guarantee some persistence properties in the simulation of a system of biochemical reactions, independently of the kinetics.

We have described a Boolean model for the problem of enumerating all minimal siphons in a Petri net and have compared two Boolean methods to the state-of-the-art algorithm from the Petri net community [11]. On the benchmark of 345 biological models from the curated part of the BioModels repository, the Boolean method for enumerating all minimal siphons using MiniSAT is very efficient. It also scales very well in the size of the net. The CLP(B) program also solves all but one instances of the benchmark, with a better

performance than MiniSAT in average, but does not scale-up as well on the largest size Petri nets, such as for instance on Kohn's map with 509 species and 775 reactions. The MiniSAT solver and the CLP(B) program both outperform the dedicated algorithms of the Petri net wommunity by one or two orders of magnitude and solve instances out of reach of these other algorithms.

The efficiency of the MiniSAT and CLP(B) methods for enumerating all solutions of an NP-complete problem for all, including large, instances of our practical benchmarks was quite surprising and lead us to study the theoretical complexity of these problems. Besides the proofs of NP-completeness of the existence of a siphon of a given cardinality, and of co-NP-completeness of the siphon-trap property, we have shown that these decision problems are tractable in Petri nets of bounded tree-width. Then we have shown that the BioModels benchmark of large biochemical networks have indeed a relatively small tree-widths.

These various results militate for the analysis of biochemical networks with Petri net concepts and Constraint Programming tools.

Acknowledgements This work was supported by the French OSEO project Biointelligence.

References

1. R. Backofen, S. Will, and E. Bornberg-Bauer. Application of constraint programming techniques for structure prediction of lattice proteins with extended alphabets. *Bioinformatics*, 15(3):234–242, 1999.
2. M. R. Birtwistle, M. Hatakeyama, N. Yumoto, B. A. Ogunnaike, J. B. Hoek, and B. N. Kholodenko. Ligand-dependent responses of the ErbB signaling network: experimental and modeling analysis. *Molecular Systems Biology*, 3(144), Sept. 2007.
3. A. Bockmayr and A. Courtois. Using hybrid concurrent constraint programming to model dynamic biological systems. In *Proceedings of ICLP'02, International Conference on Logic Programming*, volume 2401 of *Lecture Notes in Computer Science*, pages 85–99, Copenhagen, 2002. Springer-Verlag.
4. L. Calzone, A. Gelay, A. Zinovyev, F. Radvanyi, and E. Barillot. A comprehensive modular map of molecular interactions in RB/E2F pathway. *Molecular Systems Biology*, 4(173), 2008.
5. N. Chabrier-Rivier, M. Chiaverini, V. Danos, F. Fages, and V. Schächter. Modeling and querying biochemical interaction networks. *Theoretical Computer Science*, 325(1):25–44, Sept. 2004.
6. F. Chu and X.-L. Xie. Deadlock analysis of petri nets using siphons and mathematical programming. *IEEE Transactions on Robotics and Automation*, 13(6):793–804, 1997.
7. F. Commoner. Deadlocks in petri nets. *Applied Data Research Inc., Wakefield, MA*, 1972.
8. F. Corblin, S. Tripodi, E. Fanchon, D. Ropers, and L. Trilling. A declarative constraint-based method for analyzing discrete genetic regulatory networks. *Biosystems*, 98(2):91–104, 2009.
9. R. Cordone, L. Ferrarini, and L. Piroddi. Characterization of minimal and basis siphons with predicate logic and binary programming. In *Proceedings of IEEE International Symposium on Computer-Aided Control System Design*, pages 193–198, 2002.
10. R. Cordone, L. Ferrarini, and L. Piroddi. Some results on the computation of minimal siphons in petri nets. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, Maui, Hawaii USA, dec 2003.

11. R. Cordone, L. Ferrarini, and L. Piroddi. Enumeration algorithms for minimal siphons in petri nets based on place constraints. *IEEE transactions on systems, man and cybernetics. Part A, Systems and humans*, 35(6):844–854, 2005.
12. B. Courcelle. The monadic second-order logic of graphs i. recognizable sets of finite graphs. *Information and Computation*, pages 12–75, 1990.
13. B. Courcelle and I. Durand. Automata for the verification of monadic second-order graph properties. *Journal of Applied Logic*, pages 368–409, 2012.
14. J. M. Crawford and L. D. Auton. Experimental results on the crossover point in satisfiability problems. In *Proceedings of the 11th National Conference on Artificial Intelligence*, pages 21–27. AAAI press, 1993.
15. L. F. de Figueiredo, S. Schuster, C. Kaleta, and D. A. Fell. Can sugars be produced from fatty acids? a test case for pathway analysis tools. *Bioinformatics*, 25(1):152–158, 2009.
16. V. Devloo, P. Hansen, and M. Labbe. Identification of all steady states in large biological systems by logical analysis. *Bulletin of Mathematical Biology*, 65:1025–1051, 2003.
17. D. Diaz and P. Codognet. Design and implementation of the GNU Prolog system. *Journal of Functional and Logic Programming*, 6, Oct. 2001.
18. P. Dittrich and P. di Fenizio. Chemical organisation theory. *Bulletin of Mathematical Biology*, 69(4):1199–1231, Apr. 2007.
19. F. Fages, S. Soliman, and R. Coolen. CLPGUI: a generic graphical user interface for constraint logic programming. *Journal of Constraints, Special Issue on User-Interaction in Constraint Satisfaction*, 9(4):241–262, Oct. 2004.
20. E. Fanchon, F. Corblin, L. Trilling, B. Hermant, , and D. Gulino. Modeling the molecular network controlling adhesion between human endothelial cells: Inference and simulation using constraint logic programming. In *CMSB'04: Proceedings of the 20 international conference on Computational Methods in Systems Biology*, pages 104–118. Springer-Verlag, 2004.
21. G. Gottlob, N. Leone, and F. Scarcello. A comparison of structural CSP decomposition methods. *Artificial Intelligence*, 124:2000, 2000.
22. R. Goud, K. van Hee, R. Post, and J. van der Werf. Petriweb: A repository for petri nets. In S. Donatelli and P. Thiagarajan, editors, *Petri Nets and Other Models of Concurrency - ICATPN 2006*, volume 4024 of *Lecture Notes in Computer Science*, pages 411–420. Springer-Verlag, 2006.
23. M. Heiner, D. Gilbert, and R. Donaldson. Petri nets for systems and synthetic biology. In M. Bernardo, P. Degano, and G. Zavattaro, editors, *8th Int. School on Formal Methods for the Design of Computer, Communication and Software Systems: Computational Systems Biology SFM'08*, volume 5016 of *Lecture Notes in Computer Science*, pages 215–264, Bertinoro, Italy, Feb. 2008. Springer-Verlag.
24. S. Helfert, A. Estevez, B. Bakker, P. Michels, and C. Clayton. Roles of triosephosphate isomerase and aerobic metabolism in trypanosoma brucei. *Biochem. J.*, 357:117–125, 2001.
25. C. Kaleta, S. Richter, and P. Dittrich. Using chemical organization theory for model checking. *Bioinformatics*, 25(15):1915–1922, 2009.
26. R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Proceedings of a Symposium on the Complexity of Computer Computations*, pages 85–103, New York, NY, 1972. IBM Research Symposia Series, Plenum Press.
27. M. Kinuyama and T. Murata. Generating siphons and traps by petri net representation of logic equations. In *Proceedings of 2th Conference of the Net Theory SIG-IECE*, pages 93–100, 1986.
28. J. Kneis, A. Langer, and P. Rossmanith. Courcelle's theorem - a game-theoretic approach. *Discrete Optimization*, 8(4):568–594, 2011.
29. K. W. Kohn. Molecular interaction map of the mammalian cell cycle control and DNA repair systems. *Molecular Biology of the Cell*, 10(8):2703–2734, Aug. 1999.
30. A. Larhlimi and A. Bockmayr. A new constraint-based description of the steady-state flux cone of metabolic networks. *Discrete Applied Mathematics*, 157(10):2257–2266, 2009. Networks in Computational Biology.

31. K. Lautenbach. Linear algebraic calculation of deadlocks and traps. In G. Voss and Rozenberg, editors, *Concurrency and Nets Advances in Petri Nets*, pages 315–336, New York, 1987. Springer-Verlag.
32. N. le Novère, B. Bornstein, A. Broicher, M. Courtot, M. Donizelli, H. Dharuri, L. Li, H. Sauro, M. Schilstra, B. Shapiro, J. L. Snoep, and M. Hucka. BioModels Database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems. *Nucleic Acid Research*, 1(34):D689–D691, Jan. 2006.
33. M. Minoux and K. Barkaoui. Deadlocks and traps in petri nets as horn-satisfiability solutions and some related polynomially solvable problems. *Discrete Applied Mathematics*, 29:195–210, 1990.
34. D. Mitchell, B. Selman, and H. Levesque. Hard and easy distributions of sat problems. In *Proceedings of the 10th National Conference on Artificial Intelligence*, pages 459–465. AAAI press, 1992.
35. T. Murata. Petri nets: properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–579, Apr. 1989.
36. F. Nabli. Finding minimal siphons as a CSP. In *CP'11: The Seventeenth International Conference on Principles and Practice of Constraint Programming, Doctoral Program*, pages 67–72, Sept. 2011.
37. F. Nabli, F. Fages, T. Martinez, and S. Soliman. A boolean model for enumerating minimal siphons and traps in petri-nets. In *Proceedings of CP'2012, 18th International Conference on Principles and Practice of Constraint Programming*, volume 7514 of *Lecture Notes in Computer Science*, pages 798–814. Springer-Verlag, Oct. 2012.
38. F. Nabli and S. Soliman. Steady-state solution of biochemical systems, beyond S-Systems via T-invariants. In P. Quaglia, editor, *CMSB'10: Proceedings of the 8th International Conference on Computational Methods in Systems Biology*, pages 14–22. CoSBi, ACM, Oct. 2010.
39. O. Oanea, H. Wimmel, and K. Wolf. New algorithms for deciding the siphon-trap property. In *PETRI NETS'10 Proceedings of the 31st international conference on Applications and Theory of Petri Nets*, pages 267–286. Springer-Verlag, 2010.
40. J. L. Peterson. *Petri Net Theory and the Modeling of Systems*. Prentice Hall, New Jersey, 1981.
41. V. N. Reddy, M. L. Mavrovouniotis, and M. N. Liebman. Petri net representations in metabolic pathways. In L. Hunter, D. B. Searls, and J. W. Shavlik, editors, *Proceedings of the 1st International Conference on Intelligent Systems for Molecular Biology (ISMB)*, pages 328–336. AAAI Press, 1993.
42. N. Robertson and P. Seymour. Graph minors. II. Algorithmic aspects of tree-width. *Journal of Algorithms*, 7(3):309–322, 1986.
43. B. Schoeberl, C. Eichler-Jonsson, E. D. Gilles, and G. Muller. Computational modeling of the dynamics of the map kinase cascade activated by surface and internalized egf receptors. *Nature Biotechnology*, 20(4):370–375, 2002.
44. S. Soliman. Invariants and other structural properties of biochemical models as a constraint satisfaction problem. *Algorithms for Molecular Biology*, 7(15), May 2012.
45. L. Stryer. *Biochemistry*. Freeman, New York, 1995.
46. S. Tanimoto, M. Yamauchi, and T. Watanabe. Finding minimal siphons in general petri nets. *IEICE Trans. on Fundamentals in Electronics, Communications and Computer Science*, pages 1817–1824, 1996.
47. A. von Kamp and S. Schuster. Metatool 5.0: fast and flexible elementary modes analysis. *Bioinformatics*, 22(15):1930–1931, 2006.
48. M. Yamauchi and T. Watanabe. Time complexity analysis of the minimal siphon extraction problem of petri nets. *EICE Trans. on Fundamentals of Electronics, Communications and Computer Sciences*, pages 2558–2565, 1999.
49. I. Zevedei-Oancea and S. Schuster. Topological analysis of metabolic networks based on petri net theory. *In Silico Biology*, 3(29), 2003.

2.4 Circuit-based Conditions for Multistationarity

The use of the stoichiometry of models goes beyond SNA. The rank of the stoichiometric matrix is, for instance, used to compute the deficiency that is the core of the CRNT and its capacity to decide about the possibility of multistationarity. Other techniques related to circuit analysis and based on Thomas' hypotheses for multistationarity however did not rely on stoichiometry directly, but only through the Jacobian matrix.

As a result, as successful as it is in the field of logical models, the theory of Thomas and the corresponding necessary conditions, though they have been proven for ODE systems corresponding to biochemical reactions, have remained unusable in practice for many years. Indeed any network containing a reversible reaction or a multi-molecular reaction, will have a positive loop —respectively mutual activation or mutual inhibition— and therefore may (or may not) exhibit multistationarity. Thomas' condition does not bring any information.

In [17] we improve the results of [60] by revisiting their proof, armed with the stoichiometric information. This permits to rule out many positive circuits in the symbolic Jacobian since they cannot produce multistationarity, thanks to stoichiometric constraints. With this more strict condition, Thomas' theory finally becomes usable for biochemical systems and comes as a complementary tool to the recent CRNT advances [22, 34]. Furthermore, taking into account the structure permits much more simple and concise proofs, avoiding complicated rewirings of the system as in [43].

This goes to show that even when mainly concerned about the continuous semantics of a reaction system, the discrete structure remains a very powerful tool for its analysis, which is one of the main motivations behind what is described in Section 4.2.

[17] Sylvain Soliman. "A stronger necessary condition for the multistationarity of chemical reaction networks". In: *Bulletin of Mathematical Biology* 75.11 (Nov. 2013), pp. 2289–2303. DOI: 10.1007/s11538-013-9893-7

Bull Math Biol (2013) 75:2289–2303
DOI 10.1007/s11538-013-9893-7

Society for Mathematical Biology

ORIGINAL ARTICLE

A Stronger Necessary Condition for the Multistationarity of Chemical Reaction Networks

Sylvain Soliman

Received: 3 May 2013 / Accepted: 25 July 2013 / Published online: 19 September 2013

Abstract Biochemical reaction networks grow bigger and bigger, fed by the high-throughput data provided by biologists and bred in open repositories of models allowing merging and evolution. Nevertheless, since the available data is still very far from permitting the identification of the increasing number of kinetic parameters of such models, the necessity of structural analyses for describing the dynamics of chemical networks appears stronger every day.

Using the structural information, notably from the stoichiometric matrix, of a biochemical reaction system, we state a more strict version of the famous Thomas' necessary condition for multistationarity. In particular, the obvious cases where Thomas' condition was trivially satisfied, mutual inhibition due to a multimolecular reaction and mutual activation due to a reversible reaction, can now easily be ruled out.

This more strict condition shall not be seen as some version of Thomas' circuit functionality for the continuous case but rather as related and complementary to the whole domain of the structural analysis of (bio)chemical reaction systems, as pioneered by the chemical reaction network theory.

Keywords Jacobian matrix · Influence graph · Feedback circuit · Multistationarity · Chemical reaction network

1 Introduction

In the last 30 years, the conjecture of Thomas (1981) on the necessary presence of a positive circuit for the occurrence of multistationarity has opened a whole field of research:

S. Soliman (✉)
Equipe Contraintes, Inria Paris-Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 Le Chesnay Cedex, France
e-mail: Sylvain.Soliman@inria.fr

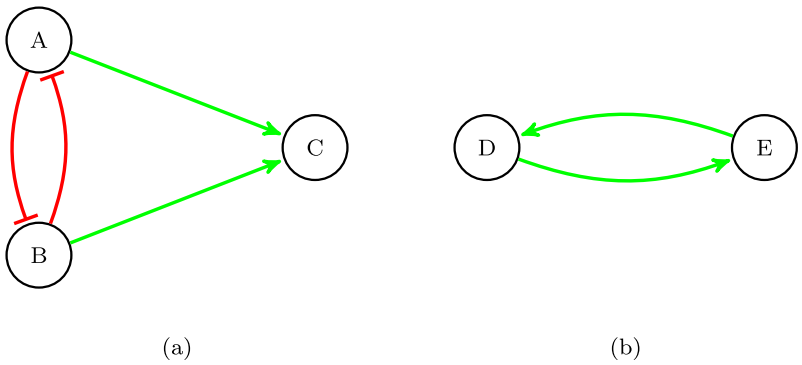


Fig. 1 (**a**): mutual inhibition resulting from `A + B => C`. (**b**): mutual activation resulting from `D <=> E`. For clarity, negative self-loops have not been represented

first, on the mathematical side, since it has been proven under various forms, depending on the restrictions on the system, discrete (Remy et al. 2008; Richard and Comet 2007) or continuous (Cinquin and Demongeot 2002; Gouzé 1998; Plahte et al. 1995; Snoussi 1998; Soulé 2003);

but also based on the insight gained from it allowing better modeling and understanding of biochemical networks, especially in the emerging field of systems biology.

However, if that second aspect is striking in the field of discrete modeling of gene regulatory networks, even the stronger versions of Thomas' necessary conditions published recently (Kaufman et al. 2007) did not have the same impact in the Ordinary Differential Equations (ODE) based modeling community.

This is mostly due to the fact that this necessary condition, the existence of a positive loop in the Jacobian of the ODE system, is almost always satisfied. Indeed, any binary reaction equipped with mass-action kinetics will lead to the mutual inhibition of the two substrates, and thus create such a loop (see Fig. 1a).

Even a reversible unary reaction can lead to satisfying the condition, since it induces a mutual activation of the two biochemical species (see Fig. 1b).

Since many models are still being constructed based on ODEs, the community turned to other types of conditions, especially for purely mass-action or non-autocatalytic (NAC) systems (Craciun and Feinberg 2005; Craciun et al. 2006).

Nevertheless, even using these conditions, Craciun and Feinberg (2006) states that enzyme kinetics of the form: `S + E <=> C => P + E` promote cycles, and thus might explain why such systems *might be far more prone than others to exhibit multiple equilibria*.

In this article, after some preliminaries, we give necessary conditions for multistationarity, not restricted to mass-action or NAC chemical reaction systems, stronger than Thomas', and local as in Soulé (2003). They show that enzyme kinetics of the above form—Fig. 1 or enzyme kinetics *à la* (Craciun and Feinberg 2006)—do not by themselves create multistationarity-inducing cycles. This is shown through a main theorem and three practical corollaries, and is extended in Sect. 5 via graph transfor-

mation operations. We then position these new results with respect to existing works and conclude on possible perspectives.

Our approach is based on the Directed Species Reaction (DSR) graph as defined by Kaltenbach (2012) (and not the homonym from Banaji and Craciun 2009) and the related analysis of terms appearing in the Jacobian's determinant. It is *local*, i.e., the existence of a positive loop can be verified for a given point of the phase space. Note that since the local version of Thomas' conjecture for oscillations has been proven false in Richard and Comet (2011), there is no direct generalization of our work to the existence of limit cycles, even using Hurwitz determinants as analogous to principal minors for Hopf bifurcations.

2 Jacobian Characterization of Multistationarity

Following Soulé (2003), we will use the Gale–Nikaido univalence theorem (Gale and Nikaido 1965) and its refinements.

We consider a differentiable map F from Ω, a product of n intervals of $\mathbb{R}$, to $\mathbb{R}^n$ and study the corresponding system $\dot{x} = F(x)$. Its *Jacobian* matrix, denoted J, is defined as usual as follows:

$$J_{ij}(x) = \partial f_i / \partial x_j(x)$$

The *influence graph* associated to J at each point x is the labeled directed graph with vertices $\{x_i \mid 1 \leq i \leq n\}$; an arc (x_i, x_j) if J_{ij} is not null, and the sign of J_{ij} as label.

The graphs depicted in Fig. 1 are influence graphs where the labels are represented as colors and arrow tips: green and pointy tip for a positive sign, red and T-tip for a negative sign.

From now on, we shall use the following terms: a *hooping* of J is a disjoint collection of cycles of its influence graph; it is *Hamiltonian* if it is covering all n vertices.

Theorem 1 (Soulé 2003) *Let F be any differentiable map from Ω to $\mathbb{R}^n$, with Jacobian matrix J. If Ω is open and F has two nondegenerate zeroes in Ω, then there exists a in Ω such that some principal minor of $-J(a)$ is negative.*

Using the Leibniz formula for determinants that defines them in terms of permutations of the indices of the matrix, and the link between those and Hamiltonian hoopings, one then obtains the classical necessary condition on the existence of some a such that there is a positive circuit in $J(a)$.

Note that this result can be extended to the case where Ω is the (closed) positive orthant, when some additional, but commonly accepted in the systems biology community, conditions are met. This is the case for instance if F is the function associated with the dynamics of a chemical reaction system with only mass-action kinetics (Craciun and Feinberg 2005), or if one is only interested in multistationarity associated to saddle-node bifurcations.

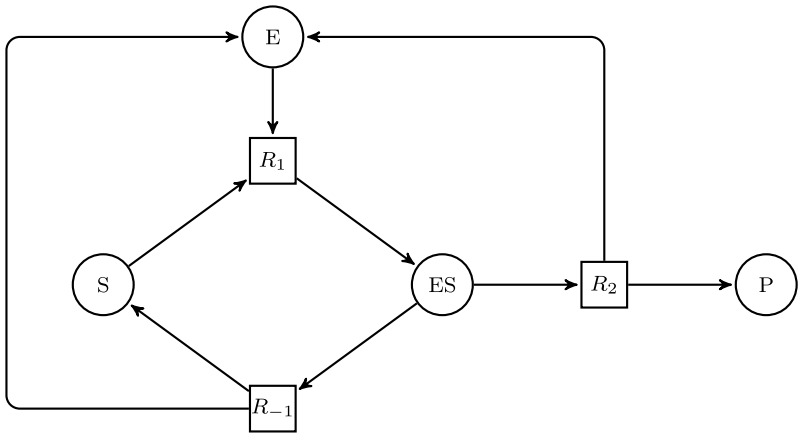


Fig. 2 Bipartite graph representation of the enzymatic reaction: `S + E <=> ES => E + P`. Arcs being all labeled with 1; this label is not shown

3 Directed Species-Reaction Graph

Let us now consider a (bio)chemical reaction system with n species $S_1, \ldots, S_n$ and m reactions $R_1, \ldots, R_m$. Using notations from Kaltenbach (2012), we write

$$R_j = \sum_{i=1}^{n} y_{ij} S_i \longrightarrow \sum_{i=1}^{n} y'_{ij} S_i$$

The y and y' represent the stoichiometric coefficients of the reactants and products of the reaction.

The rate law associated with reaction R_j will be written v_j. This defines a dynamical system in the sense of previous section: $\dot{x} = F(x)$ where x_i is the concentration of species S_i and

$$f_i(x) = \sum_j v_j(x) \cdot \left(y'_{ij} - y_{ij}\right)$$

This kind of systems encompasses most of the systems biology models developed nowadays. One can notice for instance that the Systems Biology Markup Language (SBML) (Hucka et al. 2008) can be translated to such reactions by splitting reversible reactions into forward and backward reactions and by including modifiers on both sides of the reaction (they are not affected by it, but do affect it).

Such a system can be represented naturally in a graphical form as a bipartite graph for species and reactions (Ivanova 1979; Ivanova and Tarnopolskii 1979), as depicted in Fig. 2. Arcs can also be labeled with the y and y' in a Petri-net-like manner.

Using the same bipartite vertices but different arcs and labels, it is possible to represent the Directed Species-Reaction (DSR) graph of Kaltenbach (2012). The arcs are now defined and labeled as follows:

$$\lambda(S_i, R_j) = \frac{\partial v_j}{\partial x_i}$$
$$\lambda(R_j, S_i) = y'_{ij} - y_{ij}$$

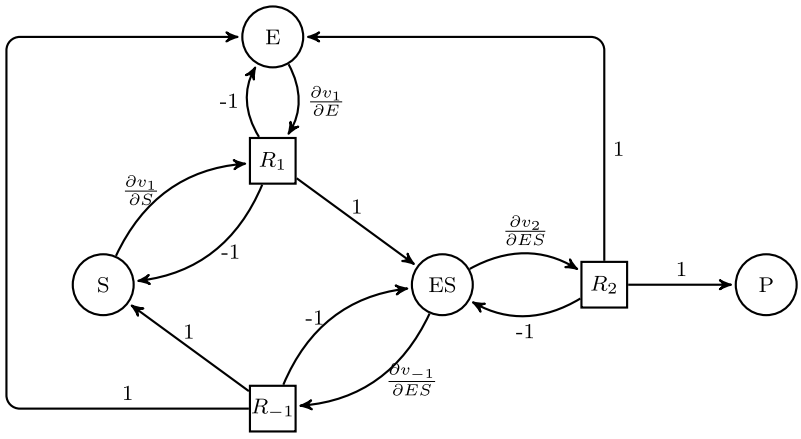


Fig. 3 DSR graph of the same enzymatic reaction: `S + E <=> ES => E + P`

Note that this is not the DSR graph defined by Banaji and Craciun (2009) since the arcs are always directed and the labels different.

If the label λ is zero, then there is no arc. λ is extended to paths (resp. subgraphs) as the product of the labels of all arcs in the path (resp. subgraph). For a path P, we shall write $\lambda_{SR}(P)$ (resp. $\lambda_{RS}(P)$) for the product of labels considering only species to reaction (resp. reaction to species) arcs.

Intuitively, the λ_{SR} represent the contribution of species to each reaction rate, whereas the λ_{RS} describe the stoichiometric effect of reactions on each species.

Figure 3 shows the DSR graph for the same chemical reaction network as Fig. 2.

Definition 1 A *species Hamiltonian hooping* of the DSR graph is a collection of cycles covering each of the species nodes exactly once.

The set of all species Hamiltonian hoopings will be denoted by $\mathcal{H}$.

Lemma 6.2 of Kaltenbach (2012) gives a decomposition of the Jacobian in terms of the set $\mathcal{H}$:

$$\det(J) = \sum_{H \in \mathcal{H}} \sigma(H)\lambda(H) \tag{1}$$

where σ is the *sign* of the species Hamiltonian hooping, defined as usual: $\sigma(H) = (-1)^{\epsilon(H)}$, where $\epsilon(H)$ denotes the number of cycles in H with an even number of species vertices. Note that, as explained in Kaltenbach (2012), this is a finer decomposition than that on the Hamiltonian hoopings of the classical influence graph, since several species-to-species paths of length two in the DSR graph might correspond to a single arc in the influence graph.

Since the DSR graph can of course be restricted to only certain species, the same lemma can be used for any principal minor of the original determinant.

Now thanks to the fact that $\lambda(H) = \lambda_{SR}(H)\lambda_{RS}(H)$, Kaltenbach (2012) groups all species Hamiltonian hoopings having the same species-to-reaction arcs using an equivalence relation noted $\sim$. We write $[H]$ for the equivalence class of a hooping H, i.e., $[H] = \{H' \in \mathcal{H} \mid H' \sim H\}$ and since $\sim$ partitions $\mathcal{H}$, we write $\mathcal{H}/\sim$ for the quotient set. One obtains the following theorem.

Theorem 2 (Kaltenbach 2012)

$$\det(J) = \sum_{[H]\in\mathcal{H}/\sim} \Lambda\big([H]\big)\lambda_{\mathrm{SR}}(H)$$

where Λ is defined as

$$\Lambda\big([H]\big) = \sum_{H'\in[H]} \sigma\big(H'\big)\lambda_{\mathrm{RS}}\big(H'\big)$$

It is important to notice that, in its original formulation, this theorem has a second part and is only stated for systems such that $\forall i, j, \partial v_i/\partial x_j \geq 0$, i.e., the λ_{SR} are always nonnegative. These systems are called NAC (non-autocatalytic) by Kaltenbach, but the hypothesis does not match the stronger and more usual definition of NAC systems (see, for instance, Banaji et al. 2007) that forbids the production of some species to depend on its concentration. It rather matches the definition of *monotonicity* (Fages and Soliman 2008a, 2008b) and allows the restriction of the sign of an element in the above sum to the different components of Λ.

Nevertheless, since we will only use the decomposition given above and whose proof does not rely on this hypothesis, we will use the theorem in its full generality.

4 Strengthening Thomas' Conjecture

Considering Theorem 1 and applying Theorem 2 to each sub-DSR-graph corresponding to a principal minor of $-J$, one can see that a necessary condition for multistationarity is that some term of the sum is negative. This again states the usual condition about the existence of a positive cycle in the influence graph of J.

We will now examine the consequences of the decomposition of this sum more precisely. In particular, we will show that many negative terms can actually be proven to cancel out with other positive terms, leaving us with more specific negative terms (and thus positive circuits) to look for.

Definition 2 The restriction of the system to a species hooping H (noted $|_H$) is the system where reactions $\{R_i \mid i \in I\}$ not appearing in H are omitted.

Since each species appears at most once in any hooping, there is one reaction associated to each species by following the outgoing arc of that species. Note, however, that the same reaction can appear several times, as associated with several different species.

This definition is naturally extended to any function or graph defined by the biochemical system.

As noticed by Kaltenbach in the proof of one of the lemmas preceding Theorem 2, Λ can be computed directly from the stoichiometric matrix $Y' - Y$ of the biochemical system. This observation is useful in proving our first lemma.

Lemma 1 *Let H be a species Hamiltonian hooping, if $(Y' - Y)|_H$ is not of full rank, then $\Lambda([H]) = 0$.*

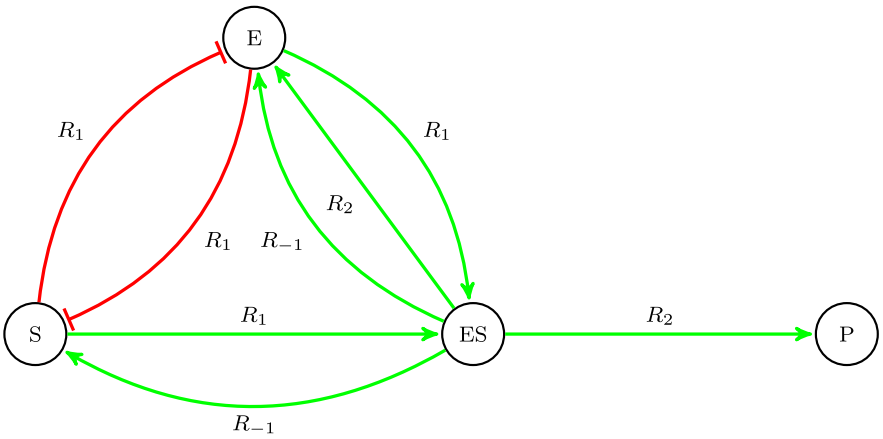


Fig. 4 Influence graph of `S + E <=> ES => E + P`. Arcs are labeled by their sign, as usual, but also by the unique reaction involved in order to obtain precisely the same species–species paths as in Fig. 3. Note for instance that there are two positive arcs from `ES` to `E`. Negative self-loops are omitted for clarity as in Fig. 1

Proof As per Lemma 6.6 of Kaltenbach (2012), we have $\Lambda([H])$ proportional to $\det((Y' - Y)|_H)$. Now, if that matrix is not of full rank, its determinant is 0. □

Remark that one can augment the usual labeling of the influence graph to contain not only the sign, but also the reaction used for each arc. There is thus an arc in this reaction-labeled influence graph for each species-to-species path of length two in the DSR graph. This leads to a one-to-one correspondence between Hamiltonian hoopings of the reaction-labeled influence graph and species Hamiltonian hoopings of the DSR graph. Figure 4 demonstrates this on the same example as Fig. 2.

In order to compute the signs of the arcs in this reaction-labeled influence graph, one now needs the sign of $\partial v_i/\partial x_j$ instead of that of $\partial f_i/\partial x_j$. However, even without precise kinetic values, this can be easily computed for most commonly used kinetics; see, for instance, Fages and Soliman (2008a) for the cases of mass action, Michaelis–Menten and Hill kinetics.

This allows us to state our main results.

Theorem 3 *Let F be any differentiable map from Ω to $\mathbb{R}^n$ corresponding to a biochemical reaction system. If Ω is open and F has two nondegenerate zeroes in Ω then there exists some a in Ω such that*:

1. *The reaction-labeled influence graph G of F at point a contains a positive circuit C;*
2. *There exists a hooping H in G, such that C is subcycle of H with $(Y' - Y)|_H$ of full rank.*

Proof Because we share the same hypotheses, we can use Theorem 1 to obtain a and the corresponding negative principal minor. It is then possible to apply Theorem 2 to decompose that minor according to the DSR graph as a sum of terms for each species Hamiltonian hooping equivalence class, and this sum must contain a negative term.

Since only positive circuits will lead to negative terms in the usual determinant decomposition, we can now prove the proposition *ab absurdum*. If all negative terms appear only in some $\Lambda([H])$ such that the restriction of the stoichiometric matrix to H is not of full rank, they will be canceled out by other terms, following Lemma 1. This would lead to a contradiction. □

Note that these properties can be checked directly on the reaction-labeled influence graph and the stoichiometry matrix.

Remark that H might not be *Hamiltonian* since it is Hamiltonian in a subgraph corresponding to the principal minor that is negative. See the end of Sect. 6 for cases when the number of species in H can be known beforehand.

Let us now apply this theorem to some common cases.

Corollary 1 *A necessary condition for the multistationarity of a biochemical reaction system is that there exists a positive cycle in its influence graph, using* at most once each reaction.

Proof This is a direct consequence of Theorem 3 since a matrix with two identical rows is clearly singular. □

It is immediate to check that the mutual inhibition resulting from bimolecular reactions—like that between `E` and `S` in our running example or between `A` and `B` in Fig. 1—cannot fulfill these necessary conditions, since the same reaction—R_1 in Fig. 4—will be repeated twice.

Corollary 2 *A necessary condition for the multistationarity of a biochemical reaction system is that there exists a positive cycle in its influence graph, not using* both forward and backward directions *of any reversible reaction*.

Proof This is once again an immediate consequence of Theorem 3 since a matrix with two opposite rows is clearly singular. □

One can thus remark that the mutual activation resulting from reversible reactions—like that between `ES` and `S` through R_1 and R_{-1} in our running example or between `D` and `E` in Fig. 1—cannot fulfill these necessary conditions.

Note that this corollary corresponds to Lemma 6.9 from Kaltenbach (2012), but with a much simpler proof involving no rewiring of the influence graph.

Other information that can be extracted from the stoichiometry is the (structural) conservation laws, i.e., P-invariants of the underlying Petri net, or more simply the left kernel of the stoichiometry matrix. Finding all the conservation laws of a biochemical model might be computationally expensive, though in practice that does not seem to be the case (Soliman 2012), but checking if a given set of species are such that their sum is constant is trivial. Based on this observation, one can state the following other corollary.

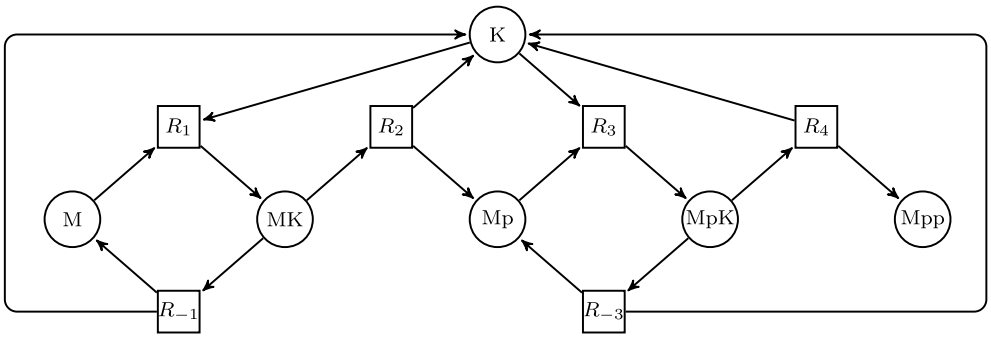


Fig. 5 Bipartite graph representation of the reactions `M + K <=> MK => K + Mp <=> MpK => K + Mpp`. Arcs being all labeled with 1; this label is not shown

Corollary 3 *A necessary condition for the multistationarity of a biochemical reaction system is that there exists a positive cycle in its influence graph, not using* all species involved in a conservation law.

Proof This is once again an immediate consequence of Theorem 3 since a weighted sum of the columns corresponding to the conservation law is zero. Therefore, a matrix containing those columns, even if some rows are forgotten via the selection of some principal minor, will be singular. □

One can thus remark that the mutual activation between `E` and `ES` through R_1 and R_2 in our running example cannot fulfill the necessary conditions for multistationarity, since `E` and `ES` form a conservation law.

The three Corollaries 1, 2, and 3 do rule out most of the obvious cases for which Thomas' condition is trivially satisfied. In particular, they show that none of the positive circuits of Fig. 4 fulfill our stricter condition.

5 Going Further

Let us now consider a two-step version of our running example, as is often considered in MAPK cascades and similar signaling pathways.

In this example, as before, most positive circuits can be proven not to be enough to make multistationarity possible. Some come from reversible reactions like R_1 and R_{-1}; some are mutual inhibitions from a single reaction like R_3; some do contain a conservation law, like the big circuit of all forms of the kinase `K` using R_1, R_2, R_3, and R_4.

However, some circuits cannot be ruled out that easily. For instance, the mutual activation of `K` and `MK` through R_1 and R_2 is now acceptable. Indeed, in the principal minor where only these two species and `Mp` are considered, the Hamiltonian hooping built from the said circuit and the self negative loop on `Mp` satisfies the hypotheses of Theorem 3.

We present here two ways to rule out this case also by transforming the system without changing the number of steady states, as noted by Soulé (2003).

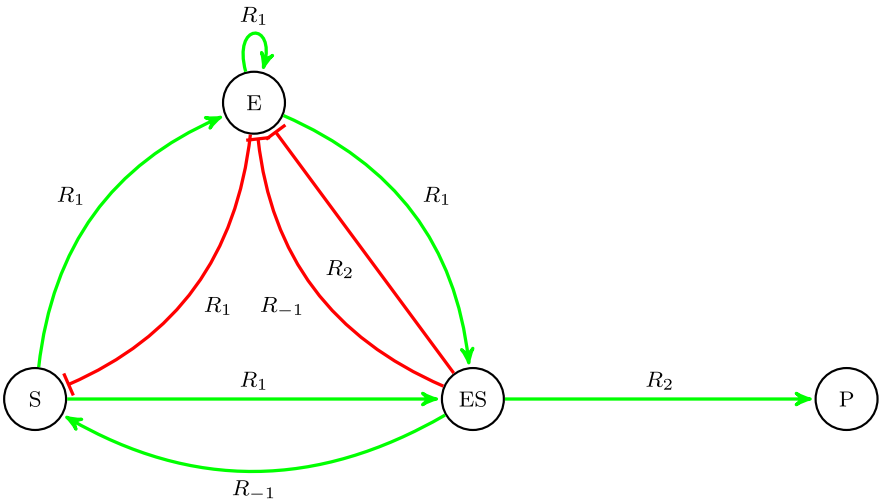


Fig. 6 Influence graph of Fig. 4 but with arcs ending in $\{E\}$ inversed. Note that the self-loop on E has become positive. All other positive cycles contain either twice R_1 or both R_1 and R_{-1}

5.1 Changing the Sign of Some f_i

First, one can multiply by -1 some of the f_i without affecting the number of steady states of the system. Let us denote by I the subset of $1, \ldots, n$ containing such indices, i.e., I is the set of indices for which the sign is changed.

This corresponds to a transformation of our reaction system where for each species of I, we exchange its stoichiometry as reactant and as product in each reaction, without modifying the rate of the reactions. In mathematical terms, for all i in I we exchange all y_{ij} and y'_{ij}, but all v_j remain untouched. This is possible since in Theorem 2 we took care not to use any hypothesis on the v_j.

The resulting labeled reaction graph is the same as before, but with the color (i.e., sign) of the arcs ending in species of I reversed. Any such graph should actually fulfill our conditions in order for the original system to be able to produce multistationarity.

Corollary 4 *A necessary condition for the multistationarity of a biochemical reaction system is that there exist positive cycles fulfilling condition* 2 *of Theorem* 3 *in the influence graph corresponding to its Jacobian*, and *in any graph obtained from it choosing a set of species and by reversing the sign of all arcs that have as target some species belonging to that set.*

Figure 6 shows the result of choosing $I = \{E\}$ in our initial example. It should be noted that there is now a positive loop on E since it had a negative loop beforehand. When trying to prove that a system cannot exhibit multistationarity, it might thus be worth restricting the search for an I to sets that contain at least some of the vertices belonging to the positive circuit that satisfies our hypotheses in the initial graph, and to sets that do not contain any vertex that had a negative self-loop in the initial graph.

In the new example, since both K and MK have negative self-loops, Corollary 4 cannot rule out the circuit.

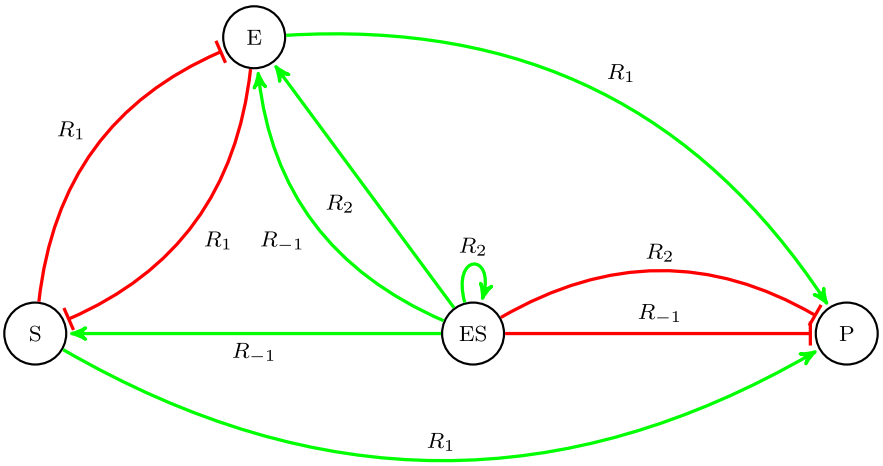


Fig. 7 Influence graph of Fig. 4 but with arcs ending in P and ES substituted. Resulting negative self-loops are, as usual, not shown, but note that the negative arcs from ES to P corresponds to negative self-loops on ES in the original graph

5.2 Permuting the Indices of Some f_i

Another transformation of the system that does not change its number of steady states is to multiply the Jacobian by a permutation, i.e., to permute the indices of some f_i.

In biochemical system terms, this can be done by applying the permutation to the species appearing in the reactions (as products or reactants), and thus to the y_{ij} and y'_{ij} but not to the rates v_j.

This transforms only the species-to-reaction arcs of the DSR graph, rewiring them according to the permutation. In the labeled influence graph, the signs and reaction labels do not change, but the arcs are rewired such that their target corresponds to the image by the permutation. All such graphs should once again fulfill our conditions in order for the original system to be able to exhibit multistationarity.

Corollary 5 *A necessary condition for the multistationarity of a biochemical reaction system is that there exist positive cycles fulfilling condition* 2 *of Theorem* 3 *in the influence graph corresponding to its Jacobian*, and *in any graph obtained from it by choosing a permutation of the species and by rewiring the arcs' target according to the permutation.*

Figure 7 shows the result of choosing a permutation of ES and P in our initial example. It should be noted that there is now a positive loop on ES since there was a positive arc from it to P beforehand. When trying to prove that a system cannot exhibit multistationarity, it might thus be worth restricting the search for a permutation to those that do not map a vertex to any other vertex such that the first one has a positive arc to it.

Actually, as noted in Soulé (2003), the two above corollaries can be combined.

Figure 8 shows that by choosing the permutation of K and Mpp in our new example, and by changing the sign on K there is no more positive cycle belonging to a hooping of full rank. This proves that the system cannot exhibit multistationarity.

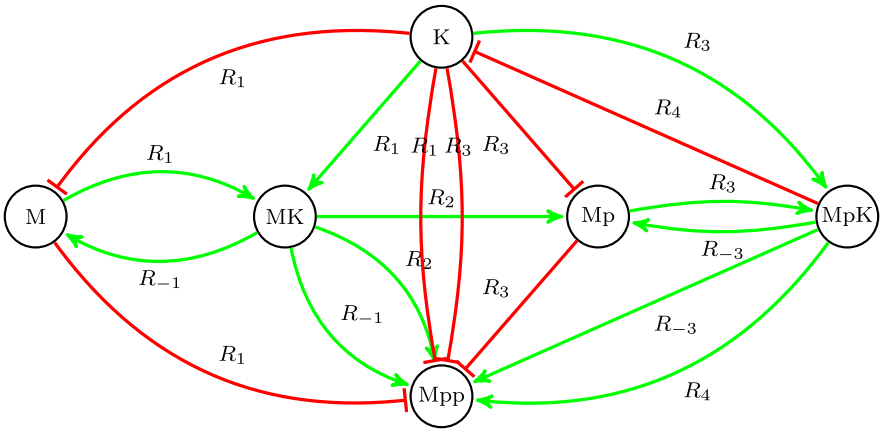


Fig. 8 Influence graph of Fig. 5, but with arcs ending in `K` and `Mpp` substituted, and then the only arc ending in `K` changed of sign. Resulting negative self-loops are, as usual, not shown, but note that the negative arcs from `K` to `Mpp` corresponds to the negative self-loops on `K` in the original graph. All positive circuits remaining can be ruled out by one of our corollaries as being a possible source of multistationarity

6 Related Work

The results we present in this paper bear some similarity with many existing works from the chemical reaction network theory community, in particular, the works of Banaji (2012) and Craciun and Feinberg (2005) that tackle multistationarity. Note, however, that the methods are different (even if another homonymous DSR graph is used) and particularly the hypotheses on the biochemical networks handled. Even with the recent works (Feliu and Wiuf 2012) lifting the requirement for having outflows for each species, the results usually require mass-action kinetics, or at least monotone rates and no auto-catalysis (i.e., NAC kinetics, see also the discussion after Theorem 2). Note that this kind of restriction forbids all the graph transformations in Sect. 5. Moreover, the conclusions are different: While the methods we present here do not handle all the examples of Craciun et al. (2006), conversely their method does not provide any conclusion for our latest MAPK example, since the *c-pair* `K Mp` from R_2 is split between the two *even-cycles* R_1, R_2 and R_2, R_3. The approaches are thus complementary, even when restricting the results of the current paper to specific types of kinetic rates.

There has been an important amount of work from Angeli and Sontag about graphical requirements in chemical reaction networks. However, as they state themselves in Angeli et al. (2010) about the work of Craciun and Feinberg, *"While we present global stability results, that work is concerned with the problem of identifying which chemical reaction networks may have multiple equilibria in their stoichiometry classes."* Indeed, they characterize monotonicity, persistence, and convergence properties of systems. These different properties in turn rely on different hypotheses than ours: they require once again NAC kinetics, whereas the present article does not.

Finally, a result that might appear similar to Theorem 3, but restricted to mass-action systems is the second theorem of Mincheva and Roussel (2007). Note, however, that their result is about the *order*, i.e., number of species, of the Hamiltonian hooping, implying that when their hypotheses are satisfied, the H we look for in

Theorem 3 covers r species, where r is the rank of the Jacobian. Their result also requires the explicit computation of K_{S_r}, the coefficient (similar to our Λ) of their critical fragment (similar to our equivalence classes over hoopings). In contrast, we provide graphical conditions for which we know from the stoichiometry—and without any hypothesis on the kinetics—that this coefficient will actually be 0, and thus that some other positive cycle should be found. Finally, once again, because of the restrictions put on the kinetic rates, their approach cannot benefit from the graph transformations presented in Sect. 5. The later article by Mincheva and Craciun (2008) does go beyond mass-action kinetics, and considers a multigraph of influences that bears some similarities with our labeled influence graph. However, by keeping only signs as labels, that approach cannot use directly results like Corollary 2 on reversible reactions.

7 Conclusion

Using the structural information, notably from the stoichiometric matrix, of a biochemical system, we have been able to state a more strict version of the famous Thomas' necessary condition for multistationarity.

Of course, since we have made no hypothesis on the kinetic rate functions, one can easily represent any dynamical system as such biochemical reactions. Typically, one would then have one reaction for each variable with rate corresponding to its derivative. In this case, the supplementary hypothesis of our theorems collapse, leaving us with the usual conditions. However, for more usual biochemical systems, it brings a concrete difference, as illustrated by Corollaries 1, 2, or 3, and if necessary their application to many transforms of the influence graph as seen in Corollaries 4 and 5.

In particular, the obvious cases where Thomas' condition was trivially satisfied, as illustrated by Figs. 1 and 4, can now easily be ruled out.

The results are local (i.e., the cycle does exist for some a in the phase space) since they rely on local theorems as was the case in Soulé (2003). Since the other Thomas' condition, on oscillations, is not (Richard and Comet 2011), one cannot directly apply the same reasoning we used to Hurwitz determinants. It might still be worth investigating, for instance when the signs of the influences in the labeled influence graph are known to be constant.

The same locality argument makes it difficult to interpret our stronger condition as some version of Thomas' circuit functionality for the continuous case. It is rather related and complementary to the whole domain of the structural analysis of (bio)chemical reaction systems, as pioneered by chemical reaction network theory. As such, we believe that our stronger necessary condition is enough to make circuit analysis a more worthwhile tool in the arsenal of the structural analyst of biochemical systems.

Acknowledgements We would like to thank Steven Gay, a Ph.D. student in our team, who was the first to formulate the conjecture proven in this article, as well as its variant on oscillations. We also thank Christophe Soulé for very interesting discussions that have led to Sect. 5. Finally, we thank the whole Contraintes team for numerous and fruitful discussions on this topic.

References

Angeli, D., De Leenheer, P., & Sontag, E. (2010). Graph-theoretic characterizations of monotonicity of chemical networks in reaction coordinates. *J. Math. Biol.*, *61*(4), 581–616.

Banaji, M. (2012). Cycle structure in sr and dsr graphs: implications for multiple equilibria and stable oscillation in chemical reaction networks. In *Transactions on Petri nets and other models of concurrency V* (pp. 1–21). Berlin: Springer.

Banaji, M., & Craciun, G. (2009). Graph-theoretic approaches to injectivity and multiple equilibria in systems of interacting elements. *Commun. Math. Sci.*, *7*(4), 867–900.

Banaji, M., Donnell, P., & Baigent, S. (2007). P matrix properties, injectivity, and stability in chemical reaction systems. *SIAM J. Appl. Math.*, *67*(6), 1523–1547.

Cinquin, O., & Demongeot, J. (2002). Positive and negative feedback: striking a balance between necessary antagonists. *J. Theor. Biol.*, *216*(2), 229–241.

Craciun, G., & Feinberg, M. (2005). Multiple equilibria in complex chemical reaction networks: I. The injectivity property. *SIAM J. Appl. Math.*, *65*(5), 1526–1546.

Craciun, G., & Feinberg, M. (2006). Multiple equilibria in complex chemical reaction networks: II. The species-reaction graph. *SIAM J. Appl. Math.*, *66*(4), 1321–1338.

Craciun, G., Tang, Y., & Feinberg, M. (2006). Understanding bistability in complex enzyme-driven reaction networks. *Proc. Natl. Acad. Sci. USA*, *103*(23), 8697–8702.

Fages, F., & Soliman, S. (2008a). Abstract interpretation and types for systems biology. *Theor. Comput. Sci.*, *403*(1), 52–70.

Fages, F., & Soliman, S. (2008b). From reaction models to influence graphs and back: a theorem. In *Lecture notes in computer science: Vol. 5054. Proceedings of formal methods in systems biology FMSB'08*. Berlin: Springer.

Feliu, E., & Wiuf, C. (2012). Preclusion of switch behavior in reaction networks with mass-action kinetics. *Appl. Math. Comput.*, *219*(4), 1449–1467.

Gale, D., & Nikaido, H. (1965). The Jacobian matrix and global univalence of mappings. *Math. Ann.*, *159*, 81–93.

Gouzé, J.-L. (1998). Positive and negative circuits in dynamical systems. *J. Biol. Syst.*, *6*, 11–15.

Hucka, M., Hoops, S., Keating, S. M., Le Novère, N., Sahle, S., & Wilkinson, D. (2008). Systems biology markup language (SBML) level 2: structures and facilities for model definitions. *Nat. Preced.* doi:10.1038/npre.2008.2715.1

Ivanova, A. N. (1979). Conditions for the uniqueness of the stationary states of kinetic systems, connected with the structures of their reaction mechanisms. *Kinet. Katal.*, *20*, 1019–1023.

Ivanova, A. N., & Tarnopolskii, B. L. (1979). One approach to the determination of a number of qualitative features in the behavior of kinetic systems, and realization of this approach in a computer (critical conditions, autooscillations). *Kinet. Katal.*, *20*, 1541–1548.

Kaltenbach, H.-M. (2012). A unified view on bipartite species-reaction and interaction graphs for chemical reaction networks. arXiv:1210.0320.

Kaufman, M., Soulé, C., & Thomas, R. (2007). A new necessary condition on interaction graphs for multistationarity. *J. Theor. Biol.*, *248*, 675–685.

Mincheva, M., & Craciun, G. (2008). Multigraph conditions for multistability, oscillations and pattern formation in biochemical reaction networks. *Proc. IEEE*, *96*(8), 1281–1291.

Mincheva, M., & Roussel, M. R. (2007). Graph-theoretic methods for the analysis of chemical and biochemical networks. I. Multistability and oscillations in ordinary differential equation models. *J. Math. Biol.*, *55*(1), 61–86.

Plahte, E., Mestl, T., & Omholt, S. W. (1995). Feedback loops, stability and multistationarity in dynamical systems. *J. Biol. Syst.*, *3*(02), 409–413.

Remy, E., Ruet, P., & Thieffry, D. (2008). Graphic requirements for multistability and attractive cycles in a boolean dynamical framework. *Adv. Appl. Math.*, *41*(3), 335–350.

Richard, A., & Comet, J.-P. (2007). Necessary conditions for multistationarity in discrete dynamical systems. *Discrete Appl. Math.*, *155*, 2403–2413.

Richard, A., & Comet, J.-P. (2011). Stable periodicity and negative circuits in differential systems. *J. Math. Biol.*, *63*(3), 593–600.

Snoussi, E. H. (1998). Necessary conditions for multistationarity and stable periodicity. *J. Biol. Syst.*, *6*, 3–9.

Soliman, S. (2012). Invariants and other structural properties of biochemical models as a constraint satisfaction problem. *Algorithms Mol. Biol.*, *7*(15). doi:10.1186/1748-7188-7-15

Soulé, C. (2003). Graphic requirements for multistationarity. *Complexus*, *1*, 123–133.
Thomas, R. (1981). On the relation between the logical structure of systems and their ability to generate multiple steady states or sustained oscillations. In *Springer Series in Synergetics* (Vol. 9, pp. 180–193).

I know that you believe that you understood what you think I said, but I am not sure you realize that what you heard is not what I meant.

Robert McCloskey, State Department spokesman

Chapter 3

Structural and Dynamical Model Reduction

Summary

3.1 Context

In recent years, the models built by biologists and modellers have grown bigger and bigger. In order to be able to use those models, not only as static knowledge repositories, but through analysis methods, for instance to identify potential drug targets, the question of model reduction has become increasingly crucial.

Furthermore, as stated about our BioIntelligence OSEO project, lead by Dassault Systèmes, the objective of pharmaceutical companies is now to see biochemical models in a product lifecycle management (PLM) perspective:

> [The BioIntelligence project] will remove several technological obstacles in order to devise a way of representing biological knowledge that is compatible with the BioPLM approach and to develop tools for systemic modelling and simulation of biological data. Other bioinformatic software developers will benefit from access to the open, integrated BioPLM software platform which will be created as a result of the programme so that they can use it to integrate their proprietary applications.

This results in the necessity to follow the evolution of a model over time, to relate it to its "ancestors", to combine it with other models without redundancy, to identify common parts, etc.

In this chapter, the first two articles focus on the notion of subgraph epimorphism (SEPI), as a systematic tool to reduce and relate models and as a more generic graph problem:

[10] Steven Gay, Sylvain Soliman, and François Fages. "A Graphical Method for Reducing and Relating Models in Systems Biology". In: *Bioinformatics* 26.18 (2010). special issue ECCB'10, pp. i575–i581. DOI: 10.1093/bioinformatics/btq388

[9] Steven Gay, François Fages, Thierry Martinez, Sylvain Soliman, and Christine Solnon. "On the subgraph Epimorphism Problem". In: *Discrete Applied Mathematics* 162 (Jan. 2014), pp. 214–228. DOI: 10.1016/j.dam.2013.08.008

The third one explores a more dynamical perspective, taking into account the kinetic expressions. It is based on the computation of tropical equilibrations for reducing quantitative models (this is an extended version of [18]). Since this technique relies on computing conservation laws, it also requires/benefits from our structural knowledge. In the long run, we hope to obtain through this kind of study conditions for the soundness and completeness of SEPI-based reductions.

[19] Sylvain Soliman, François Fages, and Ovidiu Radulescu. "A constraint solving approach to model reduction by tropical equilibration". In: *Algorithms for Molecular Biology* 9.24 (Dec. 2014). ISSN: 1748-7188. DOI: 10.1186/s13015-014-0024-2

3.2 Subgraph Epimorphisms

As striking examples, big models with almost only structural information have recently become much more common, e.g., the comprehensive map of the RB/E2F pathway compiled by Curie Institute [24], containing 530 reactions and 390 species, later on merged with the EGFR map of the Systems Biology Institute [51] and its 219 reactions and 322 species. The Reactome database (www.reactome.org) also contains several models with hundreds of molecules and reactions. Moreover, one can cite a few years old conversation on the Systems Biology Markup Language (SBML) [42] mailing list[1] giving, as biggest systems formalized so far in SBML, a large structural yeast model with 2153 species (1,168 metabolites, 832 genes, 888 proteins and 96 catalytic protein complexes) and 1857 reactions (1,761 metabolic reactions and 96 complex formation reactions) [41], obtained from merging smaller models, but with no repeated parts, or the biggest Computableplant model with 4139 reactions and 1265 species[2] or even the biggest MOOSE (multi-scale) model [30] with about 7500 species and 10000 reactions.

Once again, as in Chapter 2, we build on the structure of the models to relate them one to the other, but also to extract common parts or combine without repetition.

[1] http://sbml.org/Forums/index.php?t=tree&th=1354&mid=5041&rid=0
[2] http://computableplant.caltech.edu/models/Activator/index.html

3.2.1 A Graphical Method for Reducing and Relating Models

Completely abstracting away the kinetics might sound like an extreme way to handle model relation and reduction, however, as shown in this experiment on the whole BioModels repository, the structure of SBML models is enough to cluster them into meaningful hierarchies.

[10] Steven Gay, Sylvain Soliman, and François Fages. "A Graphical Method for Reducing and Relating Models in Systems Biology". In: *Bioinformatics* 26.18 (2010). special issue ECCB'10, pp. i575–i581. DOI: 10.1093/bioinformatics/btq388

BIOINFORMATICS
Vol. 26 ECCB 2010, pages i575–i581
doi:10.1093/bioinformatics/btq388

A graphical method for reducing and relating models in systems biology

Steven Gay, Sylvain Soliman and François Fages*
EPI Contraintes, Institut National de Recherche en Informatique et Automatique, INRIA Paris-Rocquencourt, France

ABSTRACT

Motivation: In Systems Biology, an increasing collection of models of various biological processes is currently developed and made available in publicly accessible repositories, such as biomodels.net for instance, through common exchange formats such as SBML. To date, however, there is no general method to relate different models to each other by abstraction or reduction relationships, and this task is left to the modeler for re-using and coupling models. In mathematical biology, model reduction techniques have been studied for a long time, mainly in the case where a model exhibits different time scales, or different spatial phases, which can be analyzed separately. These techniques are however far too restrictive to be applied on a large scale in systems biology, and do not take into account abstractions other than time or phase decompositions. Our purpose here is to propose a general computational method for relating models together, by considering primarily the structure of the interactions and abstracting from their dynamics in a first step.

Results: We present a graph-theoretic formalism with node merge and delete operations, in which model reductions can be studied as graph matching problems. From this setting, we derive an algorithm for deciding whether there exists a reduction from one model to another, and evaluate it on the computation of the reduction relations between all SBML models of the biomodels.net repository. In particular, in the case of the numerous models of MAPK signalling, and of the circadian clock, biologically meaningful mappings between models of each class are automatically inferred from the structure of the interactions. We conclude on the generality of our graphical method, on its limits with respect to the representation of the structure of the interactions in SBML, and on some perspectives for dealing with the dynamics.

Availability: The algorithms described in this article are implemented in the open-source software modeling platform BIOCHAM available at http://contraintes.inria.fr/biocham The models used in the experiments are available from http://www.biomodels.net/

Contact: francois.fages@inria.fr

1 INTRODUCTION

1.1 Systems biology models

Biologists use diagrams to represent interactions between molecular species. On the computer, diagrammatic notations like the Systems Biology Graphical Notation (SBGN; le Novere *et al.*, 2009) or the one introduced in Kohn's map (Kohn, 1999) of the cell cycle are also employed in interactive maps like MIM (http://discover.nci.nih.gov/mim/) (Kohn *et al.*, 2006) for instance. This kind of graphical notation encompasses two types of information: interactions (binding, complexation, protein modification, etc.) and regulations (of an interaction or of a transcription). Based on these structures, mathematical models are developed by equipping such molecular interaction networks with kinetic expressions leading to quantitative models of mainly two kinds: ordinary differential equations and continuous-time Markov chains for a stochastic interpretation of the kinetics.

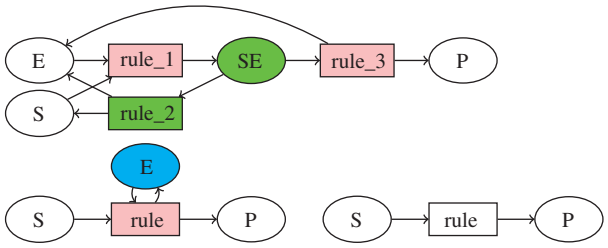


Fig. 1. Reaction graphs of the Michaelis–Menten enzymatic reaction, either complete with intermediary complex *SE*, or reduced with or without enzyme *E*. The first reduction can be achieved with the graphical operations explained in Section 2.2, for example by merging the reaction nodes *rule_1* and *rule_3* in pink into a reaction node *rule* and by deleting the green nodes *SE* and *rule_2*. The second reduction simply deletes the blue node *E*.

The Systems Biology Markup Language (SBML; Hucka *et al.*, 2003) uses a syntax of reaction rules with kinetic expressions to define such reaction models in a precise way. For instance, the Michaelis–Menten enzymatic reaction, in which an enzyme *E* transforms a substrate *S* to a product *P*, can be described either with a system of three reaction rules (equipped with mass action law kinetics) showing the formation of the intermediary complex *SE* as follows: `S + E <=> SE => P + E`, or with a single reaction rule (equipped with a Michaelis–Menten kinetics) in which the catalyst enzyme is supposed to be constant: `S + E => P + E`, and can also be omitted as in: `S => P`. These three models are represented by the bipartite graphs depicted in Figure 1, and correspond to different levels of detail for the same reaction. This is one trivial example, among others, of reduction that can be performed in large models and that we would like to identify automatically.

Nowadays, an increasing collection of models of various biological processes is indeed developed and made available to anyone in the SBML format. For instance, the publicly-accessible repository biomodels.net (le Novère *et al.*, 2006) is currently composed of 241 curated models. These different models may represent either different biological systems, or the same biological process at different levels of details or under different biological assumptions. Some represent transient directional biological processes (like signal transduction cascades), while some others represent recurrent oscillating behaviors (like circadian clock core genes or cell cycle control). Some models are

*To whom correspondence should be addressed.

pretty big (about 400 nodes, which is quite a lot for a hand-written biological model), while some others are very small (less than 10 nodes). Some models are only structural and contain only qualitative information (e.g. known protein interactions, or phenomenological events) while some others add precise quantitative data (with experiment-based kinetic rates). In some cases, the structure of the reactions is reverse-engineered from an ordinary differential equation (ODE) model and may not reflect all information, such as for instance the effect of inhibitors which cannot be distinguished from the catalysts in the syntax of a reaction rule.

1.2 Model comparison as a graph matching problem

If modelling is the process that enables our understanding and predicting of the behaviour of a system, then model reduction makes our task easier. By removing what we consider as details, model reduction allows the understanding of the core of systems, and simulation of bigger-sized systems. In mathematical biology, model reduction techniques have been studied for a long time, mainly in the case where a model exhibits different time scales, or different spatial phases, which can be analyzed separately. For instance, in the previous example of the Michaelis–Menten enzymatic reaction, the hypotheses that the substrate is in excess and the complex formation is much faster than the other reactions justify the elimination of the intermediary reactions. The mathematical conditions for quasi-steady state approximations (Segel, 1984) or total quasi-steady state approximations (Ciliberto *et al*., 2007) are however far too restrictive to be applied to Systems Biology models on a large scale, and do not take into account other abstractions than time or phase decompositions.

Our applicative purpose here is to propose a general computational method for relating models together, by considering primarily the structure of the interactions and abstracting from their dynamics and even the stoichiometry in a first step. Given two reaction graphs, the model reduction problem is to determine whether one is a reduction of the other. This model comparison focusses on the notion of model refinement that often occurs in the life-cycle of published biological models. Indeed, every biological model is 'false' at some point and can be refined to encompass more details. The modellers usually describe these refinements through two basic operations: adding new species or reactions that were unknown or considered secondary, or splitting existing species or reactions into several ones, in order to give more details (about the levels of phosphorylation of a given molecule, or about the specific mechanistic process that underlies some reaction for instance).

Graph-matching techniques have already been used for biological networks, but it is worth noticing that they have mostly been applied to either protein-interaction graphs, see for instance (Chin *et al*., 2008), or regulation graphs, see for instance (Naldi *et al*., 2009) for a dynamics-preserving graph reduction. On reaction graphs, graph-based techniques have been considered in Calzone *et al*. (2008), Radulescu *et al*. (2006) and Zinovyev *et al*. (2008) for modularization issues in large models. In this article, we study a restricted notion of subgraph epimorphism, corresponding to the application of node delete and merge operations in a reaction graph, in order to relate a source graph to a target graph through a model reduction relation.

In the next section, we present the graph-theoretic framework of model reduction by graph matching, and its formal relationship to delete and merge operations on reaction graphs. In Section 3, we describe our algorithm for solving this particular kind of graph matching problems and its implementation with a constraint program written in GNU-Prolog. Then, in Section 4, we present the graphs extracted from the biomodels repository for the evaluation, and in Section 5, we report on the performance of our algorithm and on the biological significance of the matchings found automatically in this repository. We conclude on the generality of our graphical method for model comparison, on its limits with respect to the representation of the structure of the interactions in SBML, and on some perspectives for dealing with the dynamics.

2 GRAPH MATCHING METHOD

2.1 Reaction graphs

Formally, a reaction graph G is a bipartite directed graph, that is a triple $G=(S,R,A)$, where S is the set of species nodes, R is the set of reaction nodes, and $A\subseteq S\times R\cup R\times S$ the set of arcs that describes how species interact through reactions.

There is an arc (s,r) (resp. (r,s)) if s is a reactant (resp. product) of r. Both arcs are present if s is a catalyst of r or more generally if it affects the reaction rate of r. It is worth noting that reaction graphs do not precisely model stoichiometry (hypergraphs would be needed for that) nor kinetics, but describe the structure of the interactions.

2.2 Merge and delete operations

One way to relate two models is to define graph-editing operations which make it possible to transform one reaction graph into another. A simple thing to do when trying to reduce models is to consider that two species are variants and treat them as equivalents, and to merge every interaction any of the two species had into a new species. The reaction graph formalism has a symmetry between species and reactions, so the merging process can be generalized to reactions as well, and this will prove useful.

Another natural operation is node deletion. It may be useful for instance to remove intermediate species, or species whose concentration is constant, or reactions that have become trivial after a molecular merging, or reverse reactions that occur in a much slower rate than their forward counterpart. Model refinement proceeds with the dual operations of node addition and splitting and is thus also covered by this approach.

Let us assume that $G=(S,R,A)$ is a reaction graph.

DEFINITION 2.1 (Pre/Post arcs). *Let $v\in S\cup R$, the set of* pre-arcs *(resp.* post-arcs*) of v is the set ${}^\bullet v=\{a\in A\mid\exists w\in S\cup R, a=(w,v)\}$ (resp. $v^\bullet=\{a\in A\mid\exists w\in S\cup R, a=(v,w)\}$).*

This notion extends to subsets of nodes pointwise: for $V\subseteq S$ or $V\subseteq R$ we note ${}^\bullet V=\bigcup_{v\in V}{}^\bullet v$ and $V^\bullet=\bigcup_{v\in V}v^\bullet$.

The *delete* operation removes a node from a reaction graph with all its pre- and post-arcs:

DEFINITION 2.2 (Delete). *Let $v\in S$ (resp. R), the result of the deletion of v in G is the reaction graph $d_v(G)=(S',R,A')$ (resp. (S,R',A')) where*

$$S'=S\setminus\{v\}\qquad (\text{resp. } R'=R\setminus\{v\})$$
$$A'=A\setminus(\{v\}^\bullet\cup{}^\bullet\{v\})$$

We can now define the *merge* operation that intuitively removes two vertices (either two species or two reactions) from a reaction graph and replaces them with a new one inheriting all the dangling arcs. See Figure 1 for the example of the Michaelis–Menten reduction.

DEFINITION 2.3 (Merge). *For all $v,w \in S$ (resp. R), we define $m_{v,w}(G)$ as the reaction graph (S',R,A') (resp. (S,R',A')) where*

$$S' = S \setminus \{v,w\} \uplus \{vw\} \qquad (\text{resp. } R' = R \setminus \{v,w\} \uplus \{vw\})$$

$$A' = A \setminus (\{v,w\}^\bullet \cup {}^\bullet\{v,w\}) \cup \{(vw,y) \mid (v,y) \in A \text{ or } (w,y) \in A\} \cup \{(x,vw) \mid (x,v) \in A \text{ or } (x,w) \in A\}$$

It is worth noting that these operations *delete* and *merge* for molecules and reactions can be implemented in a graphical editor for reaction rules as a mean to define model reductions, and automatically derive reduced models from simple graph editing functions. This is the case in the BIOCHAM modeling platform (Calzone *et al.*, 2006; Fages and Soliman, 2008) which now integrates novel features for editing, as well as detecting, model reductions.

2.3 Subgraph epimorphisms

DEFINITION 2.4. *Let $G=(S,R,A)$ and $G'=(S',R',A')$ be two reaction graphs. A* morphism *from G to G' is a function μ from the nodes of $G, S \cup R$, to the nodes of $G', S' \cup R'$, with $\mu(S) \subseteq S'$ and $\mu(R) \subseteq R'$, such that $\forall (x,y) \in A, (\mu(x),\mu(y)) \in A'$.*

An epimorphism *from G to G' is a morphism that is surjective on (both the nodes and the arcs of) G'. An* isomorphism *from G to G' is a morphism that is bijective on (both the nodes and the arcs of) G'.*

Notice that if there are epimorphisms from G to G' and from G' to G, then there is an isomorphism from G to G'.

As shown below, graph epimorphisms relate graphs that can be obtained by merge operations. To account for node deletions, we consider:

DEFINITION 2.5. *Let $G=(S,R,A)$ and $G'=(S',R',A')$ be two reaction graphs. A* subgraph morphism *μ from G to G' is a morphism from a subgraph induced by a subset of nodes of G, to G': $S_0 \cup R_0 \longrightarrow S' \cup R'$, $\mu(S_0) \subseteq S'$, $\mu(R_0) \subseteq R'$, with $S_0 \subseteq S$ and $R_0 \subseteq R$, such that $\forall (x,y) \in A \cap (S_0 \times R_0 \cup R_0 \times S_0), (\mu(x),\mu(y)) \in A'$.*

A subgraph epimorphism *from G to G' is a subgraph morphism that is surjective.*

In order to show the link with the *merge* and *delete* operations, we need the following properties:

LEMMA 2.6 (Commutativity). *Let $G=(S,R,A)$ be a reaction graph and $(u,v) \in S^2 \cup R^2$. $G_1 = m_{u,v}(G)$ and $G_2 = m_{v,u}(G)$ are isomorphic, i.e. there exists a bijective morphism from G_1 to G_2 (or from G_2 to G_1).*

PROOF. From Definition 2.3, it is clear that the only difference between G_1 and G_2 lies in the name of the new vertex uv or vu. The function mapping all the other vertices to themselves and uv to vu is thus a morphism from G_1 to G_2, and it is bijective.

LEMMA 2.7 (Associativity). *Let $G=(S,R,A)$ be a reaction graph and $(u,v,w) \in S^3 \cup R^3$. Then $G_1 = m_{uv,w} \circ m_{u,v}(G)$ and $G_2 = m_{u,vw} \circ m_{v,w}(G)$ are isomorphic.*

PROOF. Once again it is obvious from Definition 2.3 that both graphs have the same vertices, up to renaming of $(uv)w$ to $u(vw)$ and that these two vertices have isomorphic pre- and post-arcs corresponding to the union of all pre- and post-arcs of u, v and w. Figure 2 illustrates this.

We will denote by m_V the *merge* operation for all vertices of the set V. Notice that if V and V' are two disjoint subsets of vertices $m_V \circ m_{V'} = m_{V'} \circ m_V$. Furthermore, since $d_v \circ d_w = d_w \circ d_v$ it also makes sense to write $d_V = \bigcirc_{v \in V} d_v$.

THEOREM 2.8. *Let $G=(S,R,A)$ and $G'=(S',R',A')$ be two reaction graphs. There exists an* epimorphism *μ from G to G' if and only if there exists a finite sequence of* merge *operations, i.e. a finite sequence of pairs of vertices $(v_i,w_i)_{i \le n}$, such that the graph $m_{v_n,w_n} \circ \cdots \circ m_{v_0,w_0}(G)$ is isomorphic to G'.*

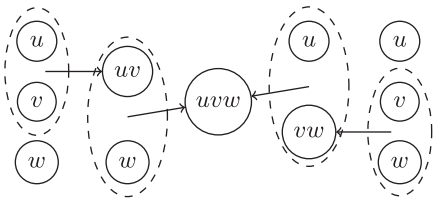


Fig. 2. Associativity of the merge operation.

PROOF. Let us prove by induction on n that if $m_{v_n,w_n} \circ \cdots \circ m_{v_0,w_0}(G)$ is isomorphic to G' then there exists an epimorphism from G to G'.

The base case is obvious since the identity is an epimorphism.

Now, suppose that $m_{v_n,w_n} \circ \cdots \circ m_{v_0,w_0}(G)$ is isomorphic to G', by induction hypothesis, there exists an epimorphism ν from G to $G'' = m_{v_{n-1},w_{n-1}} \circ \cdots \circ m_{v_0,w_0}(G)$. Now consider $\zeta : x \mapsto x$ if $x \neq v_n$ and $x \neq w_n$ and $\zeta(v_n) = \zeta(w_n) = vw$, ζ is an epimorphism from G'' to $m_{v_n,w_n}(G'')$, and thus $\mu = \zeta \circ \nu$ is an epimorphism from G to $m_{v_n,w_n} \circ \cdots \circ m_{v_0,w_0}(G)$ which is isomorphic to G'.

Conversely, suppose that μ is an epimorphism from G to G'. The set of preimages of μ partitions S and R in equivalence classes, let us write them $V_i = \mu^{-1}(v'_i)$ for $v'_i \in S' \cup R'$. Now consider $G'' = m_{V_1} \circ \cdots \circ m_{V_k}(G)$: it is isomorphic to G'. Indeed, for every i, the nodes x of V_i are merged into a single node v''_i of G'', and no V_i is empty (μ is surjective). So the function $\kappa : v'_i \longrightarrow v''_i$ is well-defined. κ is surjective on the nodes, since every node in G'' comes from the merging of a V_i, thus it is bijective on the nodes. Let $(x',y') \in A'$ Since μ is also arc-surjective, (x',y') has a preimage $(x,y) \in A$, which in turn has an image (v''_i, v''_j) in G''. So κ is a morphism. A morphism which is node-bijective is an isomorphism, hence the conclusion.

Note that this proof can actually be rephrased as a proof that sequences of merges can be associated to equivalence classes on G and then as a corrolary of the first isomorphism theorem (or of the fundamental theorem on homomorphisms).

THEOREM 2.9. *Let $G=(S,R,A)$ and $G'=(S',R',A')$ be two reaction graphs. There exists a* subgraph epimorphism *μ from G to G' if and only if there exists a finite sequence of* delete *and* merge *operations that, when applied to G, yield a graph isomorphic to G'.*

PROOF. Let us prove again the backward implication by induction on n.

The base case is still obvious since the identity is a subgraph epimorphism.

For the induction case, if the last operation is a *merge*, we obtain an epimorphism, which, composed with a subgraph epimorphism (induction hypothesis), leads to a subgraph epimorphism.

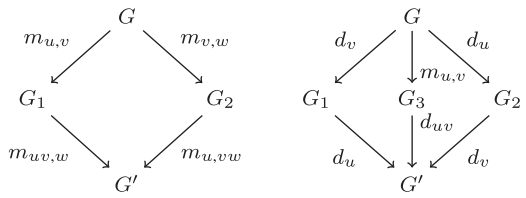
The only remaining case is when we have a subgraph epimorphism from G to G'' and G' isomorphic to $d_v(G'')$. Consider $S_0 = S \setminus \{v\}$ and $R_0 = R \setminus \{v\}$, the identity restricted to S_0 and R_0 defines a subgraph epimorphism from G'' to $d_v(G'')$, by composition we obtain a subgraph epimorphism from G to G'.

Conversely, suppose that μ is a subgraph epimorphism from G to G'. We define $S_0 = \mu^{-1}(S'), R_0 = \mu^{-1}(R')$, and writing $S' \cup R' = \{v_1,\ldots,v_n\}, V_i = \mu^{-1}(v_i)$. Now we consider $G'' = m_{V_1} \circ \cdots \circ m_{V_k} \circ d_{S \setminus S_0} \circ d_{R \setminus R_0}$: G'' is isomorphic to G' up to the renaming of the $\mu(V_i)$ by v_i. Indeed, all the $\mu(V_i)$ are different since all the V_i are disjoint, for all $(x,y) \in A \cap (S_0 \times R_0 \cup R_0 \times S_0)$ we get both an arc $(\mu(x),\mu(y))$ in A' and an arc (v_x,v_y) in G''. By definition, these are exactly the arcs of G'', and by surjectivity of μ, it also covers every arc of G'. Hence the conclusion.

Notice that if G is mapped to G' by a sequence of *merge* and *delete* operations, any sequence of merges and deletes yielding the same equivalence classes as the proof above leads to a G'' isomorphic to G'

We have seen examples of permutations between *merge* operations and between *delete* operations, another example of transformation is that of permuting a *delete* with a *merge*, one actually removes the *merge*: $d_{uv} \circ m_{u,v} = d_{\{u,v\}}$

Here are commuting diagrams summing this up:



3 ALGORITHM AND IMPLEMENTATION

The subgraph isomorphism problem is a well-known NP-complete problem, which means that there does not exist an efficient algorithm for solving *all* problem instances in polynomial time, if we admit the conjecture $P \neq NP$. Nevertheless, the practical instances of such problems may well be solved by efficient algorithms and it is the purpose of this section to describe an algorithm for our particular class of bipartite graph-matching problems. It is easy to see that our subgraph epimorphism problem is at least as hard as the graph isomorphism problem which is not known to be in *P*. However we do not know whether it is NP-complete.

The mathematical definition of subgraph epimorphisms given in the previous section can be encoded quite directly in an executable constraint program. Constraint programming is a declarative programming style which relies on two components: one modeling of the problem using elementary constraints over finite domain variables and one search procedure. Constraint programming has been applied with success to graph-matching problems in le Clément *et al*. (2009). For this work, we developed a GNU-prolog (Diaz, 2003) program dedicated to our particular subgraph epimorphism problems, using finite domain constraints and a simple search strategy for enumerating all solutions by backtracking.

Graph morphisms can be modeled quite naturally by introducing one variable per node of the source graph, with, as domain, one (integer) value per node of the target graph. A variable assignment thus represents a mapping from the source nodes to the target nodes. The morphism condition itself is written with `fd_relation` tabular constraints, which forces a tuple of variables to take its value in a list of tuples of integers.

The surjectivity property could be represented by the cardinality constraint `fd_at_least_one` of GNU-Prolog but a more efficient modeling was found by creating variables for target arcs with the set of source arcs as domain, and using the global constraint `fd_all_different`.

Then, the enumeration on the target arc variables enforces surjectivity. This enumeration is done before the enumeration of node variables that enforces the computation of a morphism.

4 DATA

The aim of our concept of subgraph epimorphism in bipartite graphs is to automatically relate and compare Systems Biology models in repositories like biomodels.net. We consider the latest version (26 January 2010) of biomodels.net which contains 241 curated models of various origins but all encoded in SBML. From the SBML format, it is possible to extract the reaction graph as follows:

(1) create a vertex for each *species*;

(2) create a vertex for each *reaction*;

(3) add an arc from a *species* to a *reaction* if it is listed in its *reactants* or *modifiers*;

(4) add an arc from a *reaction* to a *species* if it is listed in its *products* or *modifiers*.

A thematic clustering was done, using information available from the *notes* of the SBML model. We focus here on the most populated classes:

- mitogen-activated protein kinase;
- circadian clock;
- calcium oscillations;
- cell cycle.

For each class, all morphisms between pairs of models are tried.

5 RESULTS

In our algorithm, the set of all morphisms, or a proof of non-existence, are obtained by backtracking. In the experiments reported below, the computation time was limited with a timeout of 20 min but most of the problems were solved in <5 s on standard PC quadcore at 2.8 GHz.

5.1 Mapk models

The matchings found between the models of the MAPK cascade are depicted in Figure 3. This class contains the family of models of Markevich *et al*. (2005) numbered 26–31. The reductions found automatically among these models are interesting for checking whether the formalism is faithful to biological reasoning, since the authors describe refinements between them. The models are of different sizes but always consider only one level of the traditional three levels of the MAPK cascade.

In this family, models 27, 29 and 31 are the simpler ones: they have few molecules because the catalyses are represented with only one reaction. The epimorphism exhibited from model 31 to 27 corresponds to the splitting of two variants of MAPKK in 31. Model 29 distinguishes between the sites of phosphorylation of Mp, yielding a model with two molecules MpY and MpT. The subgraph epimorphism found from 29 to 27 corresponds to the deletion of one variant of Mp. Conversely, this distinction prevents the existence of an epimorphism from 31 or 27 to 29.

Models 26, 28 and 30 have more detailed catalyze mechanisms and differ as previously by the phosphorylation sites of Mp.

However, some epimorphisms from big models to small ones may have no biological meaning. This comes from the absence of constraint on the nodes that can be merged, and the relatively high number of arcs in Markevich's small models where most molecules are catalysts. Still, model 26 (with non-differentiated Mp) does not reduce to model 29 since that model indeed distinguishes MpY and MpT variants.

Now, concerning three-step MAPK cascade models, the models 9 and 11 of (Huang and Ferrell, 1996) and (Levchenko *et al*., 2000) respectively are detected as isomorphic. Indeed, they only differ by molecule names and parameter values. They do not reduce to 28 and 30, which are models that do not differentiate sites of phosphorylation. They do not reduce to 26 either, which uses a more detailed mechanism for dephosphorilations.

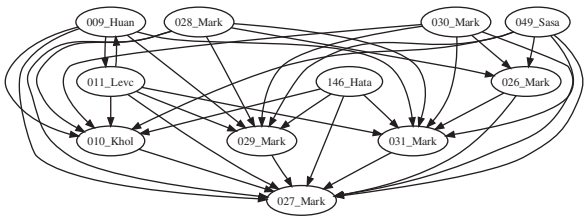


Fig. 3. Matchings found between all models of the MAPK cascade (Schoeberl's model 14 and Levchenko's model with scaffold 19 are not represented here, they do not map each other but can be mapped to small models).

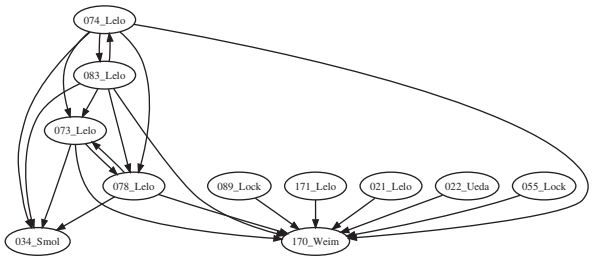


Fig. 4. Matchings between the models of the circadian clock.

Model 10 is another three-step MAPK with no catalysts for dephosphorilations. It has the particularity to be cyclic, that is, the last level's most phosphorylated molecule catalyzes the phosphorylations of the first level. This is shown here as a reduction of the previous models obtained by merging the output of the third level with the catalyst of the first level.

Finally, models 49 and 146 are bigger than the others and can easily be matched by them, and there were some comparisons for which no result was found before the timeout.

5.2 Circadian clock models

The matchings found in the class of circadian clock models are depicted in Figure 4. Models 16, 24, 25 and 36 being very small oscillators were matched by most of other models, and for that reason were left out from the picture.

Let us first have a look at the isomorphisms found.

Models 73 and 78 are isomorphic. This is in accordance with the fact that these quite detailed models come from Leloup and Goldbeter (2003) and differ indeed by parameter values.

Models 74 and 83 are isomorphic too. They also correspond to two versions of a second model from the same article, but this time with the addition of the Rev-Erbα loop, greyed out in Figure 1 of Leloup and Goldbeter (2003). The authors explain 'Taking into account explicitly the role of REV-ERBα in the indirect negative feedback exerted by BMAL1 on the expression of the Bmal1 gene requires an extension of the model, which is now governed by 19 instead of 16 kinetic equations'. The mapping to the previous models is automatically detected in accordance with these explanations, by merging the three new species (Rev-Erbα mRNA, protein in the cytoplasm and protein in the nucleus named Mr, Rc and Rn in model 74) to Bmal1 in the nucleus (named Bn in model 73).

Model 34 (Smolen *et al*., 2004) is a quite small model of the Drosophila's circadian clock. The fact that its structure is included in that of the mammalian clock of the above models is in accordance with the fact those models were built on top of knowledge from the Drosophila (Goldbeter, 1995) with a similar clock mechanism.

Models 171 (Leloup and Goldbeter, 1998) presents a model for the *Drosophila*, including Per/Tim (with two levels of phosphorylation) and the complex. Model 21 (Leloup and Goldbeter, 1999) actually studies the same model, unfortunately a different encoding in SBML (variable parameters instead of species for instance) makes it impossible to find a matching.

Many models map to model 170 (Becker-Weimann *et al*., 2004) which focusses on the positive feedback loop of the circadian cycle oscillator. It is quite small but has two compartments, which explains why only 34 cannot be reduced to it. Model 22 (Ueda *et al*., 2001) is a quite detailed model that focusses on the interlocked feedback loops, which can be mapped to 170 but not 34. Models 55 (Locke *et al*., 2005) and 89 (Locke *et al*., 2006) are both from Locke and others and about the circadian clock of *Arabidopsis* but include, in one case light induction, and in the other a new feedback loop. This explains why they do not give any matching either, except to the small oscillator model 170.

5.3 Calcium oscillation models

Figure 5 shows that many models of calcium oscillation are connected.

Models 98, 115 and 117 are in fact isomorphic due to their very small size (only two species) and differ only by their kinetics. There is a morphism from model 166 to them in accordance to the addition of a third species in this model where Ca2+ oscillations are seen as a mediator of genetic expression.

Models 43, 44 and 45 all relate to three different models from the same article (Borghans *et al*., 1997). Model 43 is the 'basic one pool' model and there is a match from 44, the '1-pool model with IP3 degradation' since the latter is indeed a refinement of the former. The morphisms from 43 and 44 to 166 correctly exhibit the inclusion of the basic three-element oscillator in those models. A false positive morphism is found however from 44 to 45, the '2-pool model'. This morphism is purely formal and has no biological meaning. It could be eliminated by using annotations as further constraints, for instance by taking into account the references to UniProt/KEGG or ChEBI databases that are already present in some SBML models.

Model 122 (Fisher *et al*., 2006) is actually a big model of NFAT and NFκB with a side calcium oscillator. However, it includes many reversible reactions and thus structurally maps to all of the other models of this class.

Model 58 is a coupled oscillator version which interestingly maps to the '2-pool' oscillator of Borghans *et al*. (1997) by merging some components of the two oscillators into one.

Finally, models 39, related to mitochondria, and 145, related to ATP-induced oscillations, only map the small oscillators already described.

5.4 Cell-cycle models

The reaction graphs of the cell cycle models are plagued by a common problem: these models originate from ODE models and the reaction graphs extracted from their encoding in SBML format does not correctly represent the structure of these models. It is thus hard to make sense of mappings between such graphs. For instance, the graphs of models 7, 8 and 56 are disconnected. Models 111, 144

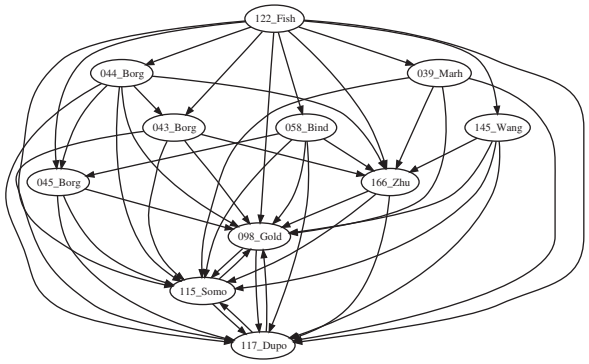


Fig. 5. Subgraph epimorphisms for models of calcium oscillations from biomodels.net.

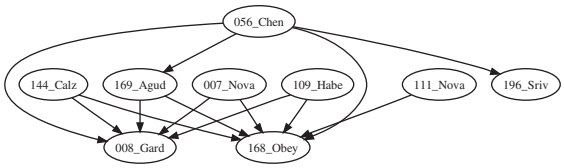


Fig. 6. SEPI for some models of the cell cycle.

and 196 have ghost molecules, that is, molecules which appear in the kinetics but not in the stoichiometry.

Nevertheless, models 144, 56 and 109 are relatively big with more than 50 reactions, and map easily on smaller models. Actually, there are 16 comparisons missing from this graph, and 13 are comparisons from these bigger graphs to the smaller ones.

Models 8, 168 and 196 are small (less than 15 reactions), which make them easy to match to, excepted for 196, which has a big diameter (Fig. 6). There is no matching from 111 to 8 however. This is explained by the erroneous structure of 8 which is disconnected.

5.5 Negative control

For the sake of completeness of the evaluation of our method, the reduction relations between all pairs of models of the biomodels.net repository have been computed (with a time out of 20 min per problem).

Some matchings between unrelated model classes were found. These false positive matchings typically arise with small models that formally appear as reductions of large models without any biological meaning, for the same reasons as in the cases discussed above within a same class. These false positives arise in less than 9% of the total inter-class pairs, and in 1.2% of the tests after the removal of the small models.

6 CONCLUSION

Constraint-based graph-matching algorithms have shown their effectiveness and efficiency to analyze and automatically relate biochemical reaction models on a large scale, namely among the 241 curated models of the systems biology repository biomodels.net. Of course, such an automatic correspondence between models inferred solely from the structure of the reaction graph may be biologically erroneous in some cases. In particular, small reaction graphs can be recognized as motifs of biologically unrelated large reaction graphs.

Nevertheless, the search for subgraph epimorphisms between all models of the biomodels.net repository revealed connected components roughly corresponding to the different models of similar biological systems for the MAPK signaling cascade, the circadian clock and the calcium oscillation models, automatically exhibiting morphisms, corresponding to model reductions, as well as isomorphisms, corresponding to variants of the same model with different parameter values.

On the other hand, the cell-cycle models of this repository often originate from ODE models that have been transcribed in SBML rules without correctly reflecting the structure of the interactions. As a result, many model reductions could not be detected as graph morphisms. More work is thus needed to curate the expression of these models in SBML, and also to restrict mappings by considering the information on molecular species present in the annotations, for instance.

Although necessarily imperfect, this approach opens a new way to query Systems Biology model repositories and study model reductions as subgraph epimorphism problems, before taking into account constraints on the stoichiometry and the dynamics of the reactions.

As a perspective for future work, the formal ground presented here in terms of graph operations and graph morphisms is currently used to investigate mathematical conditions under which the kinetics are compatible with graph reduction operations, such as for instance:

- reaction deletions for slow reverse reactions,
- reaction mergings for reaction chains with a limiting reaction,
- molecular species deletions for species in excess,
- molecular mergings for quasi-steady state approximations.

ACKNOWLEDGEMENTS

We acknowledge fruitful discussions with many participants of these projects and with Christine Solnon for an enlighting talk on graph matching at our seminar.

Funding: AE Regate; ANR Calamar (ANR-08-SYSC-003); ANR EraSysBio C5SYs; OSEO BioIntelligence projects.

Conflict of Interest: none declared.

REFERENCES

Becker-Weimann,S. *et al*. (2004) Modeling feedback loops of the mammalian circadian oscillator. *Biophys. J.*, **87**, 3023–3034.

Borghans,J.M. *et al.* (1997) Complex intracellular calcium oscillations. A theoretical exploration of possible mechanisms. *Biophy. Chem.*, **66**, 25–41.

Calzone,L. *et al*. (2006) BIOCHAM: an environment for modeling biological systems and formalizing experimental knowledge. *Bioinformatics*, **22**, 1805–1807.

Calzone,L. *et al.* (2008) A comprehensive imodular map of molecular interactions in RB/E2F pathway. *Mol. Syst. Biol.*, **4**.

Chin,G. *et al*. (2008) Biographe: high-performance bionetwork analysis using the biological graph environment. *BMC Bioinformatics*, **9** (Suppl. 6).

Ciliberto,A. *et al*. (2007) Modeling networks of coupled enzymatic reactions using the total quasi-steady state approximation. *PLOS Comput. Biol.*, **3**.

Diaz,D. (1999–2003) *GNU Prolog User'S Manual*. Available at http://www.gprolog.org/.

Fages,F. and Soliman,S. (2008) Formal cell biology in BIOCHAM. In Bernardo,M. *et al.* (eds) *8th Int. School on Formal Methods for the Design of Computer,*

Communication and Software Systems: Computational Systems Biology SFM'08, Vol. 5016 of *Lecture Notes in Computer Science*, Springer, Bertinoro, Italy, pp. 54–80.

Fisher,W.G. *et al*. (2006) Nfat and nfkappab activation in t lymphocytes: a model of differential activation of gene expression. *Ann. Biomed. Eng.*, **34**, 1712–1728.

Goldbeter,A. (1995) A model for circadian oscillations in the drosophila period protein (per). *Proc. Biol. Sci. Roy. Soc.*, **261**, 319–324.

Huang,C.-Y. and Ferrell,J.E. Jr. (1996) Ultrasensitivity in the mitogen-activated protein kinase cascade. *Proc. Natl Acad. Sci. USA*, **93**, 10078–10083.

Hucka,M. *et al*. (2003) The systems biology markup language (SBML): A medium for representation and exchange of biochemical network models. *Bioinformatics*, **19**, 524–531.

Kohn,K.W. (1999) Molecular interaction map of the mammalian cell cycle control and DNA repair systems. *Mol. Biol. Cell*, **10**, 2703–2734.

Kohn,K.W. *et al*. (2006) Molecular interaction maps of bioregulatory networks: a general rubric for systems biology. *Mol. Biol. Cell*, **17**, 1–13.

le Clément,V. *et al*. (2009) Constraint-based graph matching. In *15th International Conference on Principles and Practice of Constraint Programming (CP 2009)*, Vol. 5732 of *Lecture Notes in Computer Science*, Springer, Lisbon, Portugal, pp. 274–288.

Levchenko,A. *et al*. (2000) Scaffold proteins may iphasically affect the levels of mitogen-activated protein kinase signaling and reduce its threshold properties. *Proc. Natl Acad. Sci. USA*, **97**, 5818—5823.

le Novère,N. *et al*. (2006) BioModels Database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems. *Nucleic Acid Res.*, **1**, D689–D691.

le Novere,N. *et al*. (2009) The systems biology graphical notation. *Nat. Biotechnol.*, **27**, 735–741.

Leloup,J.-C. and Goldbeter,A. (1998) A model for circadian rhythms in drosophila incorporating the formation of a complex between the per and tim proteins. *J. Biol. Rhythms*, **13**, 70–87.

Leloup,J.-C. and Goldbeter,A. (1999) Chaos and birhythmicity in a model for circadian oscillations of the per and tim proteins in drosophila. *J. Theor. Biol.*, **198**, 445–449.

Leloup,J.-C. and Goldbeter,A. (2003) Toward a detailed computational model for the mammalian circadian clock. *Proc. Natl Acad. Sci.*, **100**, 7051–7056.

Locke,J.C. *et al*. (2006) Experimental validation of a predicted feedback loop in the multi-oscillator clock of arabidopsis thaliana. *Mol. Syst. Biol.*, **2**.

Locke,J.C. *et al*. (2005) Extension of a genetic network model by iterative experimentation and mathematical analysis. *Mol. Syst. Biol.*, **1**, msb4100018–E1–msb4100018–E9.

Markevich,N.I. *et al*. (2005) Signaling switches and bistability arising from multisite phosphorylation in protein kinase cascades. *J. Cell Biol.*, **164**, 353–359.

Naldi,A. *et al*. (2009) A reduction method for logical regulatory graphs preserving essential dynamical properties. In *CMSB'09: Proceedings of the Seventh International Conference on Computational Methods in Systems Biology*, Vol. 5688 of *Lecture Notes in BioInformatics*, Springer, Berlin, Heidelberg, pp. 266–280.

Radulescu,O. *et al*. (2006) Hierarchies and modules in complex biological systems. In Schuster,P. ed., *Proceedings of the Second European Conference on Complex Systems 2006 (ECCS 2006)*. Availabel at http://ecss.csregistry.org/ECCS%2706%20Programme.

Segel,L.A. (1984) *Modeling Dynamic Phenomena in Molecular and Cellular Biology*. Cambridge University Press, Cambridge.

Smolen,P. *et al*. (2004) Simulation of drosophila circadian oscillations, mutations, and light responses by a model with vri, pdp-1, and clk. *Biophys. J.*, **86**, 2786–2802.

Ueda,H.R. *et al*. (2001) Robust oscillations within the interlocked feedback model of drosophila circadian rhythm. *J. Theor. Biol.*, **210**, 401–406.

Zinovyev,A. *et al*. (2008) BiNoM: a cytoscape plugin for manipulating and analyzing biological networks. *Bioinformatics*, **24**, 876–877.

3.2.2 On the Subgraph Epimorphism Problem

Building on the preceding experiment, the problem of subgraph epimorphism is studied in more details, especially regarding its complexity and implementation. One important topic is the resulting definition of a graph-distance and of an "intersection" and an "union" for two input biochemical models.

These new notions fit quite well the aim of a PLM view for models by not only determining models used to build others but even common submodels or meaningful compositions of models. Unfortunately, despite our use of state of the art SAT solvers with ad-hoc encodings, solving the union and intersection problem on models of moderate size (several tens of compounds) remains very computationally expensive. A step that we plan to take, making the problem much more tractable, is definitely the use of annotations like the MIRIAM ones now added systematically to all BioModels' entries [50].

[9] Steven Gay, François Fages, Thierry Martinez, Sylvain Soliman, and Christine Solnon. "On the subgraph Epimorphism Problem". In: *Discrete Applied Mathematics* 162 (Jan. 2014), pp. 214–228. DOI: 10.1016/j.dam.2013.08.008

Discrete Applied Mathematics 162 (2014) 214–228

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

On the subgraph epimorphism problem

Steven Gay [a], François Fages [a,*], Thierry Martinez [a], Sylvain Soliman [a], Christine Solnon [b,c]

[a] EPI Contraintes, Inria Paris-Rocquencourt, Domaine de Voluceau 78150 Rocquencourt, France
[b] Université de Lyon, CNRS, France
[c] INSA-Lyon, LIRIS, UMR5205, F-69621, France

ARTICLE INFO

Article history:
Received 15 November 2011
Received in revised form 8 July 2013
Accepted 14 August 2013
Available online 17 September 2013

Keywords:
Subgraph epimorphism
Model reduction
Graph distance
Constraint solving
Systems biology

ABSTRACT

In this paper we study the problem of deciding the existence of a subgraph epimorphism between two graphs. Our interest in this variant of graph matching problem stems from the study of model reductions in systems biology, where large systems of biochemical reactions can be naturally represented by bipartite digraphs of species and reactions. In this setting, model reduction can be formalized as the existence of a sequence of vertex deletion and merge operations that transforms a first reaction graph into a second graph. This problem is in turn equivalent to the existence of a subgraph (corresponding to delete operations) epimorphism (i.e. surjective homomorphism, corresponding to merge operations) from the first graph to the second. In this paper, we study theoretical properties of subgraph epimorphisms in general directed graphs. We first characterize subgraph epimorphisms (SEPI), subgraph isomorphisms (SISO) and graph epimorphisms (EPI) in terms of graph transformation operations. Then we study the graph distance measures induced by these transformations. We show that they define metrics on graphs and compare them. On the algorithmic side, we show that the SEPI existence problem is NP-complete by reduction of SAT and present a constraint satisfaction algorithm that has been successfully used to solve practical SEPI problems on a large benchmark of reaction graphs from systems biology.

1. Introduction

Our interest in subgraph epimorphisms stems from the study of model reductions in systems biology, where large systems of biochemical reactions can be naturally represented by bipartite digraphs of species and reactions [14,10]. In this setting, one can define a very general notion of model reduction as a particular form of graph transformation and use it to compare models in systems biology model repositories [8].

Let us consider, for example, the reduction of Michaelis–Menten in Fig. 1. The left-hand side graph is a detailed model composed of three reactions where an enzyme E binds in a reversible manner to a substrate S to form a complex ES and release a product P. The right-hand side graph reduces this system to a single reaction catalyzed by the enzyme.

The reduced graph can be obtained from the source graph by a sequence of delete and merge operations on species and reaction vertices. These transformations can typically be justified in chemistry by considering: (i) reaction deletions for slow reverse reactions, (ii) reaction mergings for reaction chains with a limiting reaction, (iii) molecular species deletions for species in excess and (iv) molecular mergings for quasi-steady state approximations.

* Corresponding author. Tel.: +33 1 39 63 57 09; fax: +33 1 39 63 54 69.
E-mail addresses: Steven.Gay@inria.fr (S. Gay), Francois.Fages@inria.fr (F. Fages), Thierry.Martinez@inria.fr (T. Martinez), Sylvain.Soliman@inria.fr (S. Soliman), Christine.Solnon@liris.cnrs.fr (C. Solnon).

0166-218X/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.dam.2013.08.008

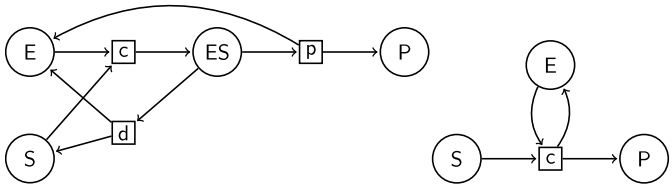


Fig. 1. A catalytic mechanism and the Michaelis–Menten reduced mechanism.

This operational view of graph reduction by graph transformation operations is equivalent to the existence of a subgraph (corresponding to delete operations) epimorphism (i.e. surjective homomorphism, corresponding to merge operations) from a source graph to a reduced graph. Subgraph epimorphisms (SEPI) differ from subgraph isomorphisms (SISO) by allowing merge operations in addition to delete operations. On undirected graphs, SEPIs differ from minors [13] with respect to the three following points: (i) non-adjacent vertices may be merged, (ii) merging adjacent vertices creates loops, and (iii) arcs cannot be deleted without deleting or merging vertices.

In this paper, we study the theoretical properties of SEPIs in general directed graphs and relate these properties with other standard notions of graph homomorphisms [9], namely subgraph isomorphisms, minors and graph epimorphisms (EPI).

Main results and overview of the paper. In Section 2, we introduce three partial orders on digraphs respectively based on SEPI , SISO and EPI , and show that, unlike the minor relation, they are not well quasi-orders. In Section 3, we introduce three graph distance measures, respectively based on SEPI , SISO and EPI , and we compare them. We show that they are metrics and that these distances are equivalent to graph edit distances defined as the minimum number of edit operations that transform a first graph into another one. In Section 5, we show the NP-completeness of the SEPI existence problem. In Section 6, we present a constraint satisfaction algorithm that has been successfully used to solve practical SEPI problems on a large benchmark of reaction graphs from systems biology. In Section 7, we discuss extensions to non -directed and bipartite graphs.

2. Partial order relations SISO , EPI and SEPI

2.1. Notations and definitions

A directed graph, or graph for short in this paper, is a pair (V, A) where V is a finite set of vertices and $A \subseteq V \times V$ a set of arcs. The cardinality of a set S is denoted as $|S|$. The size $|G|$ of a graph $G = (V, A)$ is its number of vertices, $|G| = |V|$.

For the remainder of this section, G and G' denote graphs, with $G = (V, A)$ and $G' = (V', A')$.

Definition 2.1 (*Graph Isomorphism*)**.** An isomorphism from G to G' is a bijective function $f : V \to V'$ such that $(u, v) \in A$ iff $(f(u), f(v)) \in A'$.

Two graphs G and G' are isomorphic when there exists a graph isomorphism from G to G'. Graph isomorphism is an equivalence relation on directed graphs: we note $\mathcal{G}$ the set of all graphs quotiented by this equivalence relation.

Definition 2.2 (*Graph Epimorphism*)**.** An epimorphism (EPI) from G to G' is a surjective function $f : V \to V'$ such that

- for all $u, v \in V$, if $(u, v) \in A$, then $(f(u), f(v)) \in A'$ (graph homomorphism), and,
- for all $(u', v') \in A'$, there exists $(u, v) \in A$ such that $f(u) = u'$ and $f(v) = v'$ (surjectivity on arcs).

If f is bijective, then f is a graph isomorphism. Graph epimorphisms relax the bijection constraint of graph isomorphisms to a surjection constraint on both vertices and arcs (hence the terminology of epimorphism) so that several vertices of G may be mapped on a same vertex of G'. Graph epimorphisms are closely related to graph compactions: on the class of irreflexive graphs (graphs without loops), graph epimorphisms are actually equivalent to graph compactions [19]. Graph epimorphisms are also closely related to quotient graphs (see Section 4.3).

Definition 2.3 (*Induced Subgraph*)**.** Let $U \subseteq V$ be a subset of vertices of G. The subgraph of G induced by U is $G_{\downarrow U} = (U, A \cap (U \times U))$.

Definition 2.4 (*Subgraph Isomorphism*)**.** A *subgraph isomorphism* (SISO) from G to G' is an isomorphism f from an induced subgraph G_0 of G to G'.

G_0 is the domain of f, denoted by $\mathrm{dom}\, f$.

Definition 2.5 (*Subgraph Epimorphism*)**.** A *subgraph epimorphism* (SEPI) from G to G' is an epimorphism f from an induced subgraph G_0 of G to G'.

G_0 is also denoted as $\mathrm{dom}\, f$.

Example 1. The two graphs given in the introduction for the reduction of Michaelis–Menten are related by a SEPI where the induced subgraph of the first graph is obtained by deleting the vertices *ES* and *d*, and where both vertices *c* and *p* are mapped to the vertex *c* of the second graph.

These notions thus define three relations over directed graphs, we write

- $G \overset{\text{EPI}}{\rightsquigarrow} G'$ if there exists a graph epimorphism from G to G';
- $G \overset{\text{SEPI}}{\rightsquigarrow} G'$ if there exists a subgraph epimorphism from G to G';
- $G \overset{\text{SISO}}{\rightsquigarrow} G'$ if there exists a subgraph isomorphism from G to G'.

One can easily check that the three relations $\overset{\text{SEPI}}{\rightsquigarrow}$, $\overset{\text{EPI}}{\rightsquigarrow}$ and $\overset{\text{SISO}}{\rightsquigarrow}$ are partial orders over $\mathcal{G}$.

2.2. Morphisms and graph operations

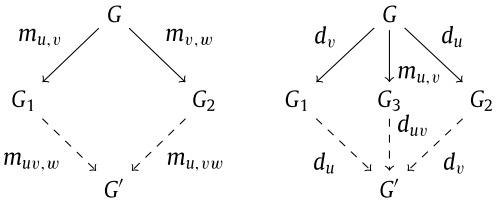
These relations are also closely related to graph transformations by delete and/or merge operations. The *delete* operation removes a vertex v from a graph G, together with every arc incident to v. In other words, it reduces G to the subgraph induced by all vertices but v.

Definition 2.6 (*Delete*)**.** Let $u \in V$. The result of the deletion of u in G is the induced subgraph $d_u(G) = G_{\downarrow V\setminus\{u\}}$.
We write $G \to_d G'$ whenever $\exists u,\ G' = d_u(G)$.

The *merge* operation removes two vertices from a graph and replaces them with a new one inheriting all incident arcs.

Definition 2.7 (*Merge*)**.** Let $u, v \in V$ such that $u \neq v$, and let uv be a new symbol such that $uv \notin V$. The result of the merge of u and v in G is the graph $m_{u,v}(G) = (V', A')$ such that $V' = V \setminus \{u, v\} \cup \{uv\}$ and $A' = A \cap (V' \times V') \cup \{(uv, w) \mid (u, w) \in A \text{ or } (v, w) \in A\} \cup \{(w, uv) \mid (w, u) \in A \text{ or } (w, v) \in A\}$.
We write $G \to_m G'$ whenever $\exists u, v,\ G' = m_{u,v}(G)$.

We have shown in [8] that these graph operations enjoy the following commutation and association properties:



These permutation properties establish the equivalence between the existence of a (sub)epimorphism from one graph to another one and the existence of a finite sequence of delete and/or merge operations leading from the first graph to the second. This is also true for subgraph isomorphisms:

Definition 2.8. We write $G \to_{md} G'$ if $G \to_d G'$ or $G \to_m G'$.
Let $o \in \{m, d, md\}$. We write $G' \leftarrow_o G$ if $G \to_o G'$, whenever it is convenient.
We write $G_1 \mathcal{R} G_2 \mathcal{R} G_3$ if $G_1 \mathcal{R} G_2 \wedge G_2 \mathcal{R} G_3$.
We write $G \mathcal{R}^* G'$ whenever there is a string of $\mathcal{R}$ relations from G to G', i.e whenever $G = G'$ or $\exists G_1 \in \mathcal{G}$ s.t. $G \mathcal{R} G_1 \mathcal{R}^* G'$.

Theorem 1 (*[8]*)**.** *$G \overset{\text{EPI}}{\rightsquigarrow} G'$ if and only if $G \to_m^* G'$.*
$G \overset{\text{SEPI}}{\rightsquigarrow} G'$ if and only if $G \to_{md}^* G'$.
$G \overset{\text{SISO}}{\rightsquigarrow} G'$ if and only if $G \to_d^* G'$.

2.3. Properties

SEPI is related to both EPI and SISO since graph epimorphisms and subgraph isomorphisms are subgraph epimorphisms, i.e., $(\overset{\text{EPI}}{\rightsquigarrow} \cup \overset{\text{SISO}}{\rightsquigarrow}) \subseteq \overset{\text{SEPI}}{\rightsquigarrow}$.

Hereditary properties have been widely studied for SISO and there exist many properties that are preserved by vertex deletions [2,3]. However, most of these properties are not preserved when considering both delete and merge operations. This comes from the fact that not only vertex mergings preserve less properties than vertex deletions, but the properties preserved by the two operations are often incompatible.

Nevertheless, one can easily check that SEPIs preserve a few graph properties.

Proposition 2. *Graph completeness is preserved by SEPIs.*

Proposition 3. *Arc symmetry is preserved by SEPIs.*

This proposition shows that SEPIs are well defined on undirected graphs.

Definition 2.9 (*Non-Neighbors*)**.** The set of outgoing (respectively incoming) *non-neighbors* of a vertex u in a graph G is the set of vertices $ONN(u, G) = \{v \in V \mid (u, v) \notin A\}$ (respectively $INN(u, G) = \{v \in V \mid (v, u) \notin A\}$).

These non-neighbor sets are monotonic with respect to SEPI :

Proposition 4. *Let f be a* SEPI *from G to G'. Then, for any vertex $x \in \operatorname{dom} f$, $f(ONN(x, G)) \supseteq ONN(f(x), G')$, and $f(INN(x, G)) \supseteq INN(f(x), G')$.*

Proof. If $ONN(x', G') = \emptyset$, the case is immediately proved.

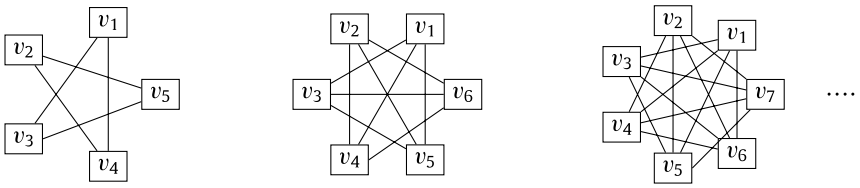
Suppose $ONN(x', G') \neq \emptyset$, and let $y' \in ONN(x', G')$. By surjectivity of f, let y such that $f(y) = y'$. We have $(x', y') \notin A'$, so, since f is a morphism, $(x, y) \notin A$. Thus $y \in ONN(x, G)$, which proves $y' \in f(ONN(x, G))$.

The proof for *INN* is similar. □

It is worth noting that SEPIs differ from minors in several ways: (i) minors allow the deletion of any arcs, whereas SEPIs only allow the deletion of vertices with their adjacent arcs; (ii) SEPIs allow the merging of non-adjacent vertices, whereas minors only allow the merging of adjacent vertices; and (iii) SEPIs create loops when merging adjacent vertices, whereas minors do not. Being a graph minor is a partial order over the set of undirected graphs and this partial order is a well-quasi-ordering [16]. This is not the case for SEPI :

Proposition 5. SEPI *is not a Well-Quasi-Order.*

Proof. One can exhibit an infinite antichain of graphs for SEPI . Let $G_n = (V_n, A_n)$ for $n \geq 5$, with $V_n = \{1 \dots n\}$ and $A_n = \{(i, j)$ such that $|i - j| > 1[n]\}$.



This family is an infinite antichain for SEPI . First, one can easily check that $\forall n, \forall u \in V_n, |ONN(x, G_n)| = 3$ (in the absence of loop a vertex is in its own ONN set). Let f be a SEPI from G_n to G_m, with $n \geq m$.

Second, f does not delete any vertex. Suppose $D = \{i \mid i \notin \operatorname{dom} f\}$ is not empty. D cannot be V_n either, or G_m would not have any vertex. So there exists i such that one vertex in $\{i, (i + 1)[n]\}$ is deleted and the other is not. Without loss of generality, say i is not deleted and $(i+1)[n]$ is. Then, by Proposition 4, $f(i)$ has at most 2 outgoing non-neighbors in G_m: itself and, if defined, the image of $(i - 1)[n]$. This is impossible, since every vertex of G_m has exactly 3 outgoing non-neighbors.

Next, f does not merge any vertices. Suppose f merges i with at least another vertex j. i and j have at most two outgoing non-neighbors in common. Indeed, let us remind that G_n is defined for $n \geq 5$: they have either two outgoing neighbors in common (when $|i-j| = 1[n]$), one (when $|i-j| = 2[n]$), or none (when $|i-j| > 2[n]$). Merging i and j makes them lose their outgoing non-neighbors not in common; thus they lose at least one outgoing non-neighbor, which is once again impossible.

Finally, f must be the identity so that $n = m$. □

Corollary 6. EPI *and* SISO *are not Well-Quasi-Orders.*

Proof. Just notice that an antichain for SEPI is also an antichain for EPI and SISO . □

3. Graph edit distance

Each of the three relations introduced in Section 2 may be used to compare some graphs. However, some other graphs cannot be compared as these relations are not total orders.

The distance between two graphs G_1 and G_2 can be defined in two main ways:

(i) in a denotational way, by means of the size of a largest subgraph common to G_1 and G_2;
(ii) in an operational way, by means of the minimum cost sequence of graph edit operations that should be performed to transform G_1 into G_2.

In [4], Bunke has connected these two definitions by introducing a special cost function for the graph edit distance and by showing that under this cost function the graph edit distance problem is equivalent to the maximum common subgraph computation.

While it is always possible to transform one graph into another by performing a sequence of vertex insertion and deletion operations, the size of such a sequence may not be representative of the similarity of the two graphs. Indeed, in some applicative contexts, it is more relevant to measure the distance between two graphs not only by means of vertex insertion and deletion operations but also by means of vertex merge and split operations, thus leading to the extended graph edit distance [1,5].

In this section, we introduce a graph edit distance corresponding to delete and merge operations. This distance is a simplified case of the extended graph edit distance introduced in [1], which has been defined for labeled graphs and is parameterized by edit costs. In the next section, we shall relate our graph edit distance to SEPI , EPI , and SISO .

Let us first define the edition graphs, which associate a vertex with every graph of $\mathcal{G}$, and an arc with every pair of graphs that can be transformed by applying one operation. Three different edition graphs can be defined according to the different kinds of operation that may be applied, i.e., m (merge), d (delete) or md (merge or delete).

Definition 3.1 (*Edition Graph*)**.** Let $o \in \{m, d, md\}$. We define the edition graph $\mathcal{E}_o = (\mathcal{G}, \mathcal{A}_o)$ such that $\mathcal{A}_o = \{(G_1, G_2) \in \mathcal{G} \times \mathcal{G} \mid G_1 \to_o G_2\}$.

These edition graphs are not strongly connected. For example, there is no path from any graph G to any graph G' that has more vertices than G, as for every arc $G_1 \to_o G_2$, we have $|G_2| = |G_1| - 1$. This prevents us from defining a metric using paths. However, there is a natural definition using the *walks* of $\mathcal{E}_o$. A path from s to t only crosses arcs forwards, walks extend paths by allowing to cross arcs forwards and backwards:

Definition 3.2 (*Walk of a Graph*)**.** Let $G = (V, A)$ be a graph, and s, t be vertices of G. A *walk* w from s to t is a finite sequence $w = (a_1 \dots a_n)$ of arcs of A such that $\exists x_0 \dots x_n \in V, x_0 = s, x_n = t,$ and $\forall i, 1 \leq i \leq n, a_i = (x_{i-1}, x_i) \vee a_i = (x_i, x_{i-1})$.
The *length* of w is $|w| = n$.

Let us now define a graph edit distance as the length of a shortest walk.

Definition 3.3 (*Distance*)**.** Let $o \in \{m, d, md\}$, and $G_1, G_2 \in \mathcal{G}$. The distance $d_o : \mathcal{G} \to \mathbb{N} \cup \{+\infty\}$ is

$$d_o(G_1, G_2) = \min\{|w| \text{ s.t. } w \text{ is a walk of } \mathcal{E}_o \text{ from } G_1 \text{ to } G_2\} \quad \text{if a walk exists}$$
$$d_o(G_1, G_2) = +\infty \quad \text{otherwise.}$$

Example 2. On the graphs G and G' of the Michaelis–Menten reduction given in the introduction, we have

$$d_{md}(G, G') = 3$$
$$d_m(G, G') = 3$$
$$d_d(G, G') = 5.$$

For $o \in \{d, md\}$, the distance $d_o(G, G')$ is never $+\infty$, as there always exists a walk from G to G' that goes through the empty graph. However, when $o = m$, it may happen that $d_o(G, G') = +\infty$. This is the case, for example, when G has no arcs whereas G' has at least one arc.

One can easily check that d_o is a metric on $\mathcal{G}$ as it satisfies the non-negativity, symmetry, separability and triangular inequality properties.

Proposition 7. *Let $G, G' \in \mathcal{G}$. Then:*

$$d_{md}(G, G') \leq d_m(G, G')$$
$$d_{md}(G, G') \leq d_d(G, G')$$
$$d_m(G, G') \leq 3d_d(G, G').$$

Proof. The two first inequalities can be proved using $\mathcal{E}_{md} \supseteq \mathcal{E}_m$ and $\mathcal{E}_{md} \supseteq \mathcal{E}_d$. The third can be proved by simulating every merge operation by the deletion of both vertices to be merged, and addition (undeletion) of the merged vertex. □

4. Relationship between d_o and EPI , SEPI and SISO

In [4], Bunke has shown that the graph edit distance that only considers vertex deletions (i.e., d_o when $o = d$) is related to the size of the maximum common subgraph. In this section, we extend this result to graph edit distances that consider vertex merges (i.e., d_o when $o \in \{m, md\}$) by relating them to EPI and SEPI .

To show this relationship, we show that for any walk w of $\mathcal{E}_o$ from G to G', there always exists a walk w' from G to G' such that $|w| \geq |w'|$ and w' changes directions at most once, i.e., it first only crosses arcs forward and then only crosses arcs backward. The vertex of $\mathcal{E}_o$ that separates forwards and backwards arc crossings corresponds to an upper bound of G and G' with respect to the partial ordering relations EPI , SEPI , or SISO .

Now, how can one compute the distance d_o between two graphs G, G'?

If $o \in \{d, md\}$, the simplest walk through the empty graph is of size $|G| + |G'|$, $d_o(G, G') \leq |G| + |G'|$, so it is sufficient to explore the graphs from size 0 to $|G| + |G'|$. When $o = d$, one can actually bound the search to graphs of the same size as $|G| + |G'|$.

It is however possible to restrict the exploration of walks to those walks that change directions at most once, by first only going down arcs and then only going up. In order to show this, we introduce quotients of graphs by equivalence relations.

4.1. Preliminaries on equivalence relations

In this section, S is a finite set. A binary relation α over S is called an *equivalence relation* over S iff it has the following properties:

- reflexivity: $\forall x \in S, (x, x) \in \alpha$
- symmetry: $\forall x \in S, \forall y \in S, (x, y) \in \alpha \Rightarrow (y, x) \in \alpha$
- transitivity: $\forall x \in S, \forall y \in S, \forall z \in S, (x, y) \in \alpha \wedge (y, z) \in \alpha \Rightarrow (x, z) \in \alpha$.

Definition 4.1. Let α be an equivalence relation over a set S and $x \in S$. The class of x modulo α, denoted by $[x]_\alpha$, is $[x]_\alpha = \{y \in A \mid (x, y) \in \alpha\}$.
The set of classes modulo α, denoted by S/α, is $S/\alpha = \{[x]_\alpha \mid x \in S\}$.

Definition 4.2 (*Transitive Closure*)**.** Let $\alpha \subseteq X \times Y$, $\beta \subseteq Y \times Z$. The *composition* of α and β is $\alpha \cdot \beta = \{(x, z) \mid \exists y, (x, y) \in \alpha \wedge (y, z) \in \beta\}$.
The *transitive closure* of α is the relation $\alpha^+ = \cup_{i=1}^{\infty} \alpha^i$ with $\alpha^1 = \alpha$ and $\forall i \geq 2, \alpha^{i+1} = \alpha \cdot \alpha^i$.
The *reflexive transitive symmetric closure* of α is the relation

$$\alpha^{\equiv} = \{(x, y) \mid (x, y) \in \alpha \vee (y, x) \in \alpha \vee x = y\}^+.$$

For any α, $\alpha^{\equiv}$ is an equivalence relation, the smallest containing α.

Definition 4.3. Let α, β be equivalence relations over S. The *product of equivalence relations* is $\alpha * \beta = (\alpha \cup \beta)^+$. It is an equivalence relation, the smallest (inclusion-wise) containing both α and β.

4.2. Dimension of an equivalence class

For the remainder of this section, let α and β be equivalence classes over S.

Definition 4.4. Let s be a binary relation over S.
s is called a *spanning* of α iff $s^{\equiv} = \alpha$.
s is called a *free family* (or just *free* for short) iff it has no loops and no cycles.

As subsets of $S \times S$, spannings are ordered by inclusion. The minimal spannings are free, analogously to minimal generating families in vector spaces. Minimal spannings share a common size, which enables the definition of dimension.

Proposition 8. *Let s be a spanning of α. Then s is minimal iff s is free.*
In this case, $|s| = |E| - |E/\alpha|$.

Proof. Suppose s is minimal and has a cycle $e_1 \dots e_n$. Since $e_n \in (s - \{e_n\})^{\equiv}$, $s - \{e_n\}$ is a spanning of α, so s is not minimal, which is absurd. Likewise, s has no loops.
Now suppose s is free. Let us prove $|s| = |E| - |E/\alpha|$. Let $n = |E/\alpha|$. Notice that $n \geq 1$. When $n = 1$, the undirected version of s ($s \cup s^{-1}$) is a tree, so if it covers $k \geq 1$ vertices, it has $k - 1$ arcs.
We have $s \subseteq \bigcup_{[x]_\alpha \in E/\alpha} [x]_\alpha \times [x]_\alpha$, so $s = \bigcup_{[x]_\alpha \in E/\alpha} s \cap ([x]_\alpha \times [x]_\alpha)$.
Since $s \cap ([x]_\alpha \times [x]_\alpha)$ is a free spanning of $[x]_\alpha \times [x]_\alpha$, the argument for $n = 1$ gives $|s \cap ([x]_\alpha \times [x]_\alpha)| = |[x]_\alpha| - 1$.
So $|s| = \sum_{[x]_\alpha \in E/\alpha} (|[x]_\alpha| - 1) = |E| - |E/\alpha|$.
To conclude, every free spanning has cardinality $|E| - |E/\alpha|$. If $s_0 \subseteq s$ with s_0 minimal, since s_0 is also free, $|s_0| = |s|$, so $s = s_0$ and s minimal. □

From Proposition 8, one can define the dimension of an equivalence relation as follows.

Definition 4.5. The *dimension* $\dim(\alpha)$ of α is the size of its minimal spanning $|E| - |E/\alpha|$.

One can then show a theorem analogous to the incomplete basis theorem.

Theorem 9. *Let $s \subseteq \alpha$ be a free family. Then there is a minimal spanning t of α that contains s.*

Proof. Let us build an increasing sequence of free families of α that contains s. Let $s_0 = s$. If s_n does not span α, then for any $(x_n, y_n) \in \alpha - s_n^{\equiv}$, $s_n \cup \{(x_n, y_n)\}$ is free. In this case we define $s_{n+1} = s_n \cup \{(x_n, y_n)\}$. This sequence grows strictly within a finite set, so it has to stop at some m. Then s_m has to span α. Since s_m is free, it is minimal. □

Next, we show that maximal free families are generating families.

Proposition 10. *Let $s \subseteq \alpha$ be free.*
If s is a maximal free family, then s is a spanning of α.
If s has size $\dim(\alpha)$, then s is a spanning of α.

Proof. For the first assertion, suppose s does not span α, that is, $\exists (x, y) \in \alpha - s^{\equiv}$. Then $x \neq y$, since $s^{\equiv}$ is reflexive. Since $s \cup \{(x, y)\}$ is not free, it must have a cycle that must contain (x, y), with every other arc of the cycle in s. This last statement means that $(x, y) \in s^{\equiv}$, which is absurd.

For the last assertion, suppose free family $s \subseteq \alpha$ has size $\dim(\alpha)$. Using Theorem 9, let t be a minimal spanning of α that contains s. Since $\dim(\alpha) = |t| \geq |s| = \dim(\alpha)$, we have $s = t$. □

Proposition 11.

$$\dim(\alpha * \beta) \leq \dim(\alpha) + \dim(\beta).$$

Proof. Let s, t be minimal spannings of α, β. $s \cup t$ is a spanning of $\alpha * \beta$, with $|s \cup t| \leq |s| + |t|$; a minimal spanning will be even smaller. □

The equality case gives an interesting result: in this case, for every s, t minimal spannings of α, β, $s \cup t$ has no cycle.

4.3. Quotients of graphs by equivalence relations

In this section, $G = (V, A)$ is a directed graph (i.e. $A \subseteq V \times V$), and α, β are equivalence classes over V.

Definition 4.6 (*Quotient Graph*)**.** The *quotient* of G by α is
$G/\alpha = (\{[x]_\alpha \mid x \in V\}, \{([x]_\alpha, [y]_\alpha) \mid (x, y) \in A\})$.

Notice that G/α has $|V/\alpha|$ vertices.
Graph epimorphisms and graph quotients are strongly linked, in the sense conveyed by the following theorem.

Theorem 12. *There exists an epimorphism f from G to G' iff there exists an equivalence relation α over V such that G/α is isomorphic to G'.*

Proof. $\Rightarrow$ Let $\alpha = \{(x, y) \in V^2 \mid f(x) = f(y)\}$. Then G/α is isomorphic to G'.
$\Leftarrow$ Let $f : V \to \mathcal{P}(V)$ such that $\forall u \in V, f(u) = [u]_\alpha$. Then f is an epimorphism from G to G/α. □

We shall use a classical theorem about equivalence classes.

Theorem 13. *Let γ be an equivalence relation with $\alpha \subseteq \gamma$. Then γ/α is an equivalence relation over V/α, and* $\dim(\gamma/\alpha) = \dim(\gamma) - \dim(\alpha)$.

Proof. Showing that γ/α is an equivalence class over V/α is easy. The result on dimensions is less well-known.
Let s be a minimal spanning of α. From Proposition 8 s is a free family of γ, so by Theorem 9, there is a $t \supseteq s$ that is a minimal spanning of γ. t being a graph, we can consider quotienting it by α. One can show that t/α is actually a spanning of γ/α.
Now t/α generally contains loops and may not be a minimal spanning. Let $t_0 = t - s$. The arcs s/α are exactly the loops of t/α, which means $(t_0/\alpha)^{\equiv} = (t/\alpha)^{\equiv} = \gamma/\alpha$.
One can prove that t_0/α is free, and its size is $|t| - |s|$, which allows us to conclude on the dimension of γ/α. □

Proposition 14. *Let γ be an equivalence relation such that $\alpha \subseteq \gamma$. Then $(G/\alpha)/(\gamma/\alpha)$ is isomorphic to G/γ.*

Proof. Notice how the vertices of G/γ are subsets of V, and the vertices of $(G/\alpha)/(\gamma/\alpha)$ are subsets of subsets of V. It is easy to check that $m : X \in V/\alpha \longrightarrow \bigcup_{x \in X} x$ is an isomorphism from the latter graph to the former. □

4.4. Graph distances and homomorphisms

We now show a practical way to compute distances d_o by restricting the search space.
Let us begin by a simple property of our operations.

Proposition 15. *If there is a sequence of merge and delete operations that transforms G into G', then this sequence has $|G| - |G'|$ operations.*

Proof. Each *merge* or *delete* operation decrements the number of vertices of G by 1. □

Let us show that in $\mathcal{E}_d$, there is always a short walk with a down–up pattern.

Proposition 16. *Let $G, G_1 = (V_1, A_1), G_2 = (V_2, A_2) \in \mathcal{G}$. If there is a walk $w = G_1 \leftarrow_d^* G \rightarrow_d^* G_2$, then there exist a graph G' and a walk $w' = G_1 \rightarrow_d^* G' \leftarrow_d^* G_2$, with $|w'| \leq |w|$.*

Proof. Let $V' = V_1 \cap V_2$, and $G' = G_{\downarrow V'}$. There is a deletion string from G_1 (respectively G_2) to G', by deleting vertices $V_1 \setminus V_2$ (respectively $V_2 \setminus V_1$).
G contains vertices $V_1 \cup V_2$, but it also has the vertices that have been deleted from both paths to G_1 and to G_2 so that $|G| \geq |V_1 \cup V_2|$.
Using Proposition 15, w has length $|G| - |G_1| + |G| - |G_2| \geq |V_1 \cup V_2| - |V_1| + |V_1 \cup V_2| - |V_2| = |V_1 \cup V_2| - |V_1 \cap V_2|$, and w' has length $|V_1| - |V_1 \cap V_2| + |V_2| - |V_1 \cap V_2| = |V_1 \cup V_2| - |V_1 \cap V_2|$.
Thus $|w'| \leq |w|$. □

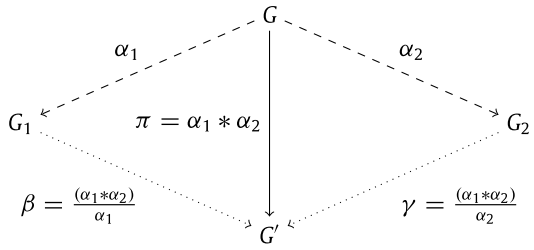
Theorem 17. *If there is a walk w of $\mathcal{E}_d$ from G_1 to G_2, then there are a graph G_c and a walk $w' = G_1 \to_d^* G_c \leftarrow_d^* G_2$ not longer than w.*

Proof. By recursion on the number of maximal 'peaks' of w, i.e. the number of maximal subwords of $w \in \to_d \cdot \leftarrow_d$: using Proposition 16 decreases the number of maximal peaks, and at each step the resulting walk from G_1 to G_2 is shorter or has the same length. □

To show the same kind of properties for merge operations, graph quotients come into play.

Proposition 18. *Let $G, G_1, G_2 \in \mathcal{G}$. If there is a walk $w = G_1 \leftarrow_m^* G \to_m^* G_2$, then there exist a graph G' and a walk $w' = G_1 \to_m^* G' \leftarrow_m^* G_2$, with $|w'| \leq |w|$.*

Proof. This illustrates the construction:



Using Theorem 12, we see a string of merge operations as one graph quotient. Let α_1 and α_2 such that $G_1 = G/\alpha_1$ and $G_2 = G/\alpha_2$. Let $\pi = \alpha_1 * \alpha_2$.

With $\beta = \pi/\alpha_1$ and $\gamma = \pi/\alpha_2$, we have, using Proposition 14, $G/\pi = G_1/\beta = G_2/\gamma$ (in other words, $\frac{G}{\alpha_1 * \alpha_2} = \frac{G}{\alpha_1}/\frac{(\alpha_1 * \alpha_2)}{\alpha_1} = \frac{G}{\alpha_2}/\frac{(\alpha_1 * \alpha_2)}{\alpha_2}$).

So, by transitivity, G' has epimorphisms from G_1 and G_3. And since β and γ have respectively smaller dimensions than α_1 and α_2 (by Proposition 11 and Theorem 13), the dotted lines in the figure above are shorter than the dashed lines, hence the property. □

Theorem 19. *If there is a walk w of $\mathcal{E}_m$ from G_1 to G_2, then there are a graph G_c and a walk $w' = G_1 \to_m^* G_c \leftarrow_m^* G_2$ not longer than w.*

Proof. Same proof as Theorem 17, using Proposition 18. □

This result works for the merge operation. However, it can be extended to the merge and delete operations at the same time. In short, vertex deletion can be simulated by merging with a dummy vertex.

Definition 4.7 (*Dummy Vertices, Pointed Graphs*)**.** Let $\bot$ be a fresh symbol that is not a vertex of any graph in $\mathcal{G}$.

Let $\cdot_\bot : G \in \mathcal{G} \longrightarrow G_\bot = (V_\bot, A_\bot)$, where $V_\bot = V \uplus \bot$ and $A_\bot = A \uplus (V \times \{\bot\}) \uplus (\{\bot\} \times V) \uplus \{(\bot, \bot)\}$.

We call the *dummy vertex* of a graph G, one that has all possible arcs to/from the other vertices of G and to itself. In $G_\bot$, $\bot$ is always a dummy vertex.

We write the set of pointed graphs $\mathcal{G}_\bot = \{G_\bot \mid G \in \mathcal{G}\}$. We extend the merge operation on $\mathcal{G}_\bot$ with no special treatment for $\bot$: a priori, the image of $\bot$ can be any vertex, and the antecedents of $\bot$ can be any vertex.

Proposition 20. *$\cdot_\bot$ is an isomorphism from $\mathcal{E}_m$ to $(\mathcal{E}_\bot)_m$:*
$G \to_{md} G'$ if and only if $G_\bot \to_m G'_\bot$.

Proof. The left to right implication is straightforward. Let $\mu : G \longrightarrow G'$ be the function corresponding to the operation, merging or deletion. If the operation is a merging, then sending $\bot$ to $\bot$ is valid. If it is a deletion, sending $\bot$ to $\bot$ and sending the deleted vertex to $\bot$ makes the operation a merging. So $\mu_\bot : \begin{pmatrix} v \in \mathrm{dom}(\mu) \longrightarrow \mu(v) \\ v \in V - \mathrm{dom}(\mu) \longrightarrow \bot \\ \bot \longrightarrow \bot \end{pmatrix}$ is a merging from $G_\bot$ to $G'_\bot$.

The converse implication should be done carefully. Let us call $\mu_\bot : G_\bot \longrightarrow G'_\bot$ the function corresponding to the merging. Notice that $\mu_\bot(\bot)$ is not necessarily $\bot$. However $\mu_\bot(\bot)$ is necessarily a dummy vertex so that $\omega : \begin{pmatrix} v \notin \{\mu_\bot(\bot), \bot\} \longrightarrow v \\ \mu_\bot(\bot) \longrightarrow \bot \\ \bot \longrightarrow \mu_\bot(\bot) \end{pmatrix}$ is a graph isomorphism of $G_\bot$.

Let $\rho = \omega \circ \mu_\bot$: it is a merging from $G_\bot$ to $G'_\bot$ with $\rho(\bot) = \bot$. One can check that $\mu : \big(v \longrightarrow \rho(v) \quad \text{if } \rho(v) \neq \bot\big)$ is either a merging (when $\rho^{-1}(\bot) = \{\bot\}$), or a deletion of u (when $\rho^{-1}(\bot) = \{\bot, u\}$). □

Theorem 21. *If there is a walk w of $\mathcal{E}_{md}$ from G_1 to G_2, then there are a graph G_c and a walk $w' = G_1 \to_{md}^* G_c \leftarrow_{md}^* G_2$ not longer than w.*

Proof. Combining Theorem 19 and Proposition 20 yields the result. □

These results on down–up walks lead to an interesting corollary when considering morphisms.

Corollary 22. *Let $G, G' \in \mathcal{G}$.*

- $d_d(G, G') = |G| + |G'| - 2\max\{|G_c| \text{ s.t. } G \overset{\text{SISO}}{\rightsquigarrow} G_c \wedge G' \overset{\text{SISO}}{\rightsquigarrow} G_c\}$
- $d_m(G, G') = |G| + |G'| - 2\max\{|G_c| \text{ s.t. } G \overset{\text{EPI}}{\rightsquigarrow} G_c \wedge G' \overset{\text{EPI}}{\rightsquigarrow} G_c\}$
- $d_{md}(G, G') = |G| + |G'| - 2\max\{|G_c| \text{ s.t. } G \overset{\text{SEPI}}{\rightsquigarrow} G_c \wedge G' \overset{\text{SEPI}}{\rightsquigarrow} G_c\}$.

Proof. First use Theorem 1 to transpose Theorems 17, 19 and 21 to morphisms, and then use Proposition 15 for cardinalities. □

The first equality is already known: the deletion distance between G and G' is the number of deletions to the maximum common induced subgraph G_c [4], which is *the* greatest lower bound of G and G' for the SISO partial order.

The other two equalities are new: G_c is a greatest lower bound of maximal cardinality for the EPI (respectively SEPI) partial orders. Note that there may be several common graphs of maximum cardinality.

Corollary 23. *$d_o(G, G')$ has the same parity as $|G| + |G'|$.*

5. Computational complexity

The SISO and EPI decision problems are NP-complete [19], and computing d_{SISO} or d_{EPI} is NP-hard since $d_o(G, G') = |G| - |G'|$ if and only if $G \overset{r}{\rightsquigarrow} G'$, by Proposition 15 and Corollary 22.

Definition 5.1. The *subgraph epimorphism problem* is the following decision problem: "*given two graphs G and G', is $G \overset{\text{SEPI}}{\rightsquigarrow} G'$ or not*?".

We prove the NP-completeness of the SEPI decision problem by the direct reduction of SAT [6]. For a finite set of variables X, let $\neg X = \{\neg x_1, \ldots, \neg x_m\}$ denotes the set of negative literals constructed upon X. For a Boolean formula in conjunctive normal form $\phi = c_1 \wedge \cdots \wedge c_n$ over X, we have for all $1 \leq i \leq n$, $c_i = \ell_{i,1} \vee \cdots \vee \ell_{i,n_i}$, and for all $1 \leq j \leq n_i$, $\ell_{i,j} \in X \cup \neg X$. Let us write $C_\phi = \{i \in \mathbb{N} \mid 1 \leq i \leq n\}$ and $L_\phi = \{(i, j) \in \mathbb{N}^2 \mid 1 \leq i \leq n, 1 \leq j \leq n_i\}$.

Lemma 24. *A Boolean formula ϕ in conjunctive normal form is satisfiable, if and only if there exists a subset $X \subseteq L_\phi$ such that*

- *for all $1 \leq i \leq n$, there exists $1 \leq j \leq n_i$ such that $(i, j) \in X$,*
- *for all $(i, j), (i', j') \in X$, $\ell_{i',j'} \neq \neg\ell_{i,j}$.*

Proof. If $\mu : V \to \{0, 1\}$ satisfies ϕ, we pose for all $x \in V$, $\mu(\neg x) = 1 - \mu(x)$ and then it suffices to observe that $X = \{(i, j) \in L \mid \mu(\ell_{i,j}) = 1\}$ satisfies the conditions of the lemma. Conversely, given a subset $X \subseteq L_\phi$ satisfying these conditions, we pose $\mu : V \to \{0, 1\}$ such that for all $x \in V$, $\mu(x) = 1$ if there exists $(i, j) \in X$ such that $\ell_{i,j} = x$ and 0 otherwise. Then, we observe that μ satisfies ϕ. □

We say that *ϕ is satisfied by X* if X is a subset of L_ϕ satisfying these conditions.

Theorem 25. *The subgraph epimorphism problem is NP-complete.*

Proof. The subgraph epimorphism problem is NP since, given two graphs $G = (V, A)$ and $G' = (V', A')$ and function $f : V \to V'$, checking whether f is a subgraph epimorphism or not can be done in polynomial time. Therefore, it suffices to show that the subgraph epimorphism problem is NP-hard.

Let us assume a SAT instance given by a Boolean formula in conjunctive normal form ϕ over a finite set of variables X. Let G and G' be the following two graphs:

$$G = (L_\phi, \{\{(i, j), (i', j')\} \subseteq L_\phi \mid i = i' \wedge j \neq j' \vee \ell_{i',j'} = \neg\ell_{i,j}\})$$
$$G' = (C_\phi, \emptyset).$$

This construction is depicted in Fig. 2.

The theorem is then an immediate consequence of the following lemma.

Lemma 26. *ϕ is satisfiable if and only if there exists a* SEPI *from G to G'.*

Suppose that ϕ is satisfied by a set $K \subseteq L_\phi$. Let $K' = \{(i, \min\{j \mid (i, j) \in K\}) \mid 1 \leq i \leq n\}$. K' is a matching, i.e. for all $i \in C_\phi$, there exists a unique j_i, with $(i, j_i) \in L_\phi$ such that $(i, j_i) \in K'$. Notice that K' still satisfies ϕ. Let $\mu : (i, j) \in K' \mapsto i \in C_\phi$. We show that μ is a SEPI from G to G'. Indeed, μ is surjective, since K' satisfies ϕ. Furthermore, the subgraph of G induced by K' has no edges: if (i, j) and (i', j') are in K, then either $i = i'$, and then $j = j'$ because K' is a matching, or $i \neq i'$, and then $\ell_{i',j'} \neq \neg\ell_{i,j}$ because $K' \subseteq K$. Since there are no edges to preserve, μ is trivially a morphism. So μ is a SEPI from G to G'.

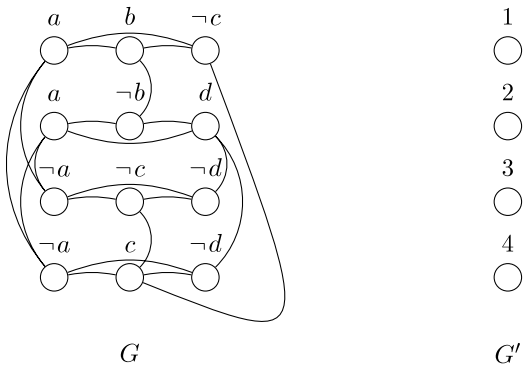


Fig. 2. Reduction from the SAT instance $\phi = (a \vee b \vee \neg c) \wedge (a \vee \neg b \vee d) \wedge (\neg a \vee \neg c \vee \neg d) \wedge (\neg a \vee c \vee \neg d)$ to SEPI.

Conversely, suppose that μ is a SEPI from G to G'. Let $K = \mu^{-1}(C_\phi)$. We show that K satisfies ϕ. First, μ is onto, so $|K| \geq n$. Then, if for some i there are distinct (i, j), $(i, j') \in K$, the edge between (i, j) and (i, j') could not be preserved by μ, so there is at most one $(i, j) \in K$ for each $i \in \{1, \ldots, n\}$. We deduce $|K| = n$, so there is exactly one $(i, j) \in K$ for each i. Furthermore, if $\ell_{i',j'} = \neg\ell_{i,j}$ for some (i, j) and $(i', j') \in K$, then the arc between the two vertices could not be preserved by μ. So K satisfies ϕ. □

It is worth noticing that in this proof, the existence of a SEPI from G to G' is equivalent to the existence of a SISO from G to G'. Therefore, this reduction shows the NP-hardness of both SEPI and SISO.

6. Constraint logic program

Despite its theoretical computational complexity, SEPI problems can be efficiently solved for some applications. In this section, we present a constraint logic program that has been successfully used to solve practical SEPI problems on a large benchmark of reaction graphs from systems biology [8]. This program implemented in GNU-Prolog [7] uses the built-in constraints `relation` and `element_var` described in the next section.

6.1. Preliminaries on constraint satisfaction problems

Definition 6.1 (*CSP*)**.** A *constraint satisfaction problem* (CSP for short) is a triple $\mathcal{P} = (V, D, C)$, where

- V is a set of variables.
- D is a family of domains indexed by variables from V : $\forall X \in V$, D_X is a finite set.
- C is a set of constraints, each $c \in C$ is defined by its arity $\text{ar}(c) \in \mathbb{N}$, a tuple of variables $\vec{X}(c) \in V^{\text{ar}(c)}$ and a relation $R(c) \subseteq \prod_{i=1}^{ar(c)} D_{\vec{X}_i(c)}$.

Definition 6.2 (*Solution of a CSP*)**.** An assignment $\eta : X \in V \longrightarrow \eta(X) \in D_X$ is a solution of $\mathcal{P}$ when $\forall c \in C$, $(\eta(X_1), \ldots, \eta(X_n)) \in R(c)$, with $\vec{X}(c) = (X_1, \ldots, X_n)$.

Graph matching problems can easily be modeled as constraint satisfaction problems [12]. One variable per vertex in the source graph is introduced; its domain is the set of vertices of the target graph. Symbolic constraints on these variables are then used to express the matching problem and to actively prune the search space by filtering the domain of these variables during search.

We shall use the following symbolic constraints and their associated domain filtering algorithms (available in GNU-Prolog [7]):

- `relation` constrains variables using a relation given in extension. The constraint $c =$ `relation`$(\vec{Y}, \mathcal{R})$, where $\vec{Y}$ is a tuple of variables and $\mathcal{R}$ is a relation, is defined by
  - $\text{ar}(c)$ = arity of $\vec{Y}$ = arity of $\mathcal{R} = n$
  - $\vec{X}(c) = \vec{Y}$
  - $R(c) = \mathcal{R} \cap \prod_{i=1}^{n} D_{Y_i}$,
- `element_var` constrains a list of variables f describing a function to have Y as the image of its X-th element. The constraint $c =$ `element_var`(X, f, Y), where $X, Y \in V$ and $f = (f_1, \ldots, f_n) \in V^n$, is defined by
  - $\text{ar}(c) = n + 2$
  - $\vec{X}(c) = (X, f_1, \ldots, f_n, Y)$
  - $R(c) = \{(index, \eta_1, \ldots, \eta_n, image) \in D_X \times (\prod_{i=1}^{n} D_{f_i}) \times D_Y \mid \eta_{index} = image\}$

6.2. SEPI as a constraint satisfaction problem

The graph epimorphism problem between two graphs G and G' can be modeled as a constraint satisfaction problem $\mathcal{P} = (V, D, C)$ as follows. Variables are associated with the vertices of G and G' and to the edges of G'

$$V = \{X_v \mid v \in V(G)\} \uplus \{Y_{v'} \mid v' \in V(G')\} \uplus \{Y_{e'} \mid e' \in E(G')\},$$

with domains respectively

$$D(X_v) = V(G'_\perp),$$
$$D(Y_{v'}) = \{1, \ldots, |V(G)|\},$$
$$D(Y_{e'}) = \{1, \ldots, |E(G)|\}.$$

The constraints are

$$\begin{aligned} C = {} & \{\texttt{relation}((X_u, X_v), E(G'_\perp)) \mid (u, v) \in E(G)\} \\ & \cup \{\texttt{element\_var}(Y_{v'}, X(V(G)), v') \mid v' \in V(G')\} \\ & \cup \{\texttt{element\_var}(Y_{(u',v')}, \pi_1(X(E(G))), u') \mid (u', v') \in E(G')\} \\ & \cup \{\texttt{element\_var}(Y_{(u',v')}, \pi_2(X(E(G))), v') \mid (u', v') \in E(G')\} \end{aligned}$$

where $X(V(G))$ is an arbitrarily ordered list containing the elements of $\{X_v \mid v \in V(G)\}$, $X(E(G))$ is a list representation of the set $\{(X_u, X_v) \mid (u, v) \in E(G)\}$, and π_1, π_2 map the first and second projection functions on lists.

Proposition 27. *The CSP problem $\mathcal{P}$ associated with graphs G, G' has a solution if and only if there exists a subgraph epimorphism from G to G'.*

Proof. We prove that a variable assignment η is a solution to $\mathcal{P}$ if and only if the restriction of η to $\{u \in V(G) \mid \eta(u) \in V(G')\} = \eta|_{V(G)}^{V(G')}$ is a subgraph epimorphism from G to G'.

If η is a solution to $\mathcal{P}$, then $\eta|_{V(G)}^{V(G')}$ preserves the arcs of G since for each $(u, v) \in E(G)$, $\texttt{relation}((X_u, X_v), E(G'))$ enforces $(\eta(X_u), \eta(X_v)) \in E(G')$. Moreover, $\eta|_{V(G)}^{V(G')}$ is surjective on vertices of G', since for each $v' \in V(G')$, $\texttt{element\_var}(Y_{v'}, X(V(G)), v')$ forces the $Y_{v'}$-th element of $V(G)$ to have v' as its image, and similarly surjectivity on arcs of G' is enforced by the remaining constraints.

Conversely, suppose that there is a subgraph epimorphism f from G to G'. Let $g : V(G') \mapsto V(G)$ be any inverse of f on the vertices (i.e. $\forall v' \in V(G') f(g(v')) = v'$), and $h : E(G') \mapsto E(G)$ any inverse of f on the arcs. Let η such that

- $\forall v \in V(G)$, $\eta(X_v) = f(v)$,
- $\forall v' \in V(G')$, $\eta(Y_{v'}) = i$ s.t. $V(G)_i = g(v')$,
- $\forall (u', v') \in E(G')$, $\eta(Y_{(u',v')}) = j$ s.t. $E(G)_j = h((u', v'))$.

Then η is a solution to the CSP $\mathcal{P}$. □

Interestingly, the $\perp$ vertices used to reduce SEPI to EPI in the proof of Proposition 20 are also used in this coding of subgraph epimorphism as a CSP: without the dummy vertices, we obtain a CSP encoding of the EPI problem.

6.3. The CSP framework: propagation and enumeration

In order to solve a real-world CSP problem, the enumeration of tuples η is not a viable possibility. Constraint solvers use the fact that from the reduction of the domain of some variables, one can deduce information about the other variables: in particular, one can deduce forbidden values for other variables.

As an example, consider the CSP with variables $\{X, Y, Z\}$ and constraints $(X, Y) \in \{(1, 1), (2, 3)\}$, $(Y, Z) \in \{(1, 2), (1, 3)\}$. Instantiating X to 1 allows the deduction of $Y = 1$, so we never need to test $Y = 2$: this deduction has been *propagated* from $X = 1$. To continue the solving process, we have no choice but to let $Y = 2$, and we can choose the next value to try for Z : $Z = 2$ is a valid choice.

If at first we had tried $X = 2$, the first constraint would have forbidden $Y = 1$, and the second constraint would have forbidden every value of Z. This is a *failure*: on failure, constraint solvers backtrack to the last instantiation $Var = val$ where there was a choice, remove val from the domain of Var, and try another value for Var. If no value remains, the backtracking process has to return to an even earlier choice point, and if there is no such choice point, then there is no solution to the CSP.

Searching for a solution by constraint solving alternates between propagation and enumeration steps. The general searching scheme is described in the following function $\textsc{Search}_C(D)$: this function returns a solution η in the domain family D if such a solution exists, and fails otherwise.

function $\textsc{Search}_C(D)$
  $D' \leftarrow \textsc{Propagate}_C(D)$
  if $\exists X \mid D'_X = \emptyset$ **then**

```
        fail
    else if ∀X, |D'_X| = 1 then
        return η such that ∀X, D'_X = {η(X)}
    else
        (X, d) ← VariableValueSelection(D')
        try
            D'' ← (X ↦ d; Y ≠ X ↦ D'_Y)
            Search_C(D'')
        on failure
            D'' ← (X ↦ D'_X \ d; Y ≠ X ↦ D'_Y)
            Search_C(D'')
        end try
    end if
end function
```

Propagation is assumed to be monotonic ($\textsc{Propagate}_C(D_X) = D'_X \subseteq D_X$ for all $X \in V$) and correct (if the condition $\forall X, |D'_X| = 1$ holds, then the resulting η is a solution). Enumeration is assumed to be strictly monotonic ($d \subsetneq D'_X$) and inhabited ($d \neq \emptyset$). The function VariableValueSelection should never select a singleton variable: VariableValueSelection $(D) = (X, d) \Rightarrow |D_X| > 1$. Therefore, $\textsc{Search}_C(D)$ terminates.

In the worst case, with no propagation ($D' = D$), the search procedure is equivalent to a non-deterministic labeling in $\mathbf{O}(d^n)$ where d is the size of the largest domain (the maximum between the number of vertices in G' and the number of arcs in G) and n is the number of variables (the sum of the number of vertices in G and arcs in G').

We consider propagation algorithms that ensure the domain–arc-consistency of the domain family $D' = \textsc{Propagate}_C(D)$ with respect to the set of constraints C. A domain family D' is domain-consistent with respect to a set of constraints C when for every variable $X \in V$ and every value $v \in D'_X$, there exists an assignment η such that $\eta(X) = v$ and η respects every constraint in C.

The built-in constraint propagators of GNU Prolog indeed maintain arc-consistency for the constraints `relation` and `element_var`.

6.4. Search strategy for SEPI

Here we discuss the choice of VariableValueSelection for SEPI CSPs.

The default choice could be to use a generic strategy for enumerating both source vertices and antecedent vertices and arcs as follows:

```
function VariableValueSelection(D)
    if ∃X | D_X = {d_1, d_2, ...} then
        return (X, d_1)
    end if
end function
```

Actually, the enumeration of only one of the sets is sufficient. Let us consider the following enumeration function:

```
function VariableValueSelection_{X(V(G))}(D)
    if ∃v ∈ V(G) | D_{X_v} = {d_1, d_2, ...} then
        return (X_v, d_1)
    else if ∃v' ∈ V(G') | Y_{v'} = {d_1, d_2, ...} then
        return (Y_{v'}, d_1)
    else if ∃(u', v') ∈ E(G') | Y_{(u',v')} = {d_1, d_2, ...} then
        return (Y_{(u',v')}, d_1)
    end if
end function
```

Proposition 28. *For a* SEPI *CSP, the search strategy with* `VariableValueSelection`$_{X(V(G))}$ *yields a solution once the source vertex variables are instantiated, for any instantiation of the antecedent variables in their domains.*

Proof. First, suppose that we have tried to enumerate the source vertex variables, and failed. Then, the correctness of constraint propagation ensures that there is no SEPI from the source graph to the target graph.

Conversely, if on the contrary the enumeration on source vertex variables succeeded, then there is obviously a morphism. Is it surjective? The domain–arc-consistency of `element_var` removes values $v \in V(G)$ (respectively $(u, v) \in E(G)$) from antecedent variables of v' (respectively (u', v')) as soon as it is known that the image of v (respectively (u, v)) is not v' (respectively (u', v')).

Since every source vertex variable is completely instantiated, the domains of antecedent variables are more than a set of possible antecedents: they are the exact sets of antecedents.

From the $\textsc{Search}_C$ algorithm, every antecedent variable has a non-empty domain, which means the morphism is surjective, and any value for the antecedent variables will satisfy the constraints. □

Let us now consider the enumeration of antecedent variables as follows:

function VARIABLEVALUESELECTION$_{antecedents}$(D)
  if $\exists v' \in V(G') \mid Y_{v'} = \{d_1, d_2, \ldots\}$ **then**
    return $(Y_{v'}, d_1)$
  else if $\exists (u', v') \in E(G') \mid Y_{(u',v')} = \{d_1, d_2, \ldots\}$ **then**
    return $(Y_{(u',v')}, d_1)$
  else if $\exists v \in V(G) \mid |D_{X_v}| > 1 \wedge D_{X_v} \neq \{\bot\}$ **then**
    return $(X_v, \bot)$
  end if
end function

Proposition 29. *For a* SEPI *CSP, the search strategy with* `VariableValueSelection`$_{antecedents}$ *yields a solution once every antecedent variable has been instantiated, by instantiating the remaining non-singleton source vertex variables to* $\bot$.

Proof. First, suppose we have enumerated only the antecedent variables, and failed. Once again, it is obvious that there is no SEPI from the source graph to the target graph.

Conversely, if the enumeration on antecedent variables succeeded, then some source vertex variables already have singleton domains. The induced subgraph formed by the source vertices that correspond to these variables is sufficient to cover G', and the correctness of the `relation` propagator ensures that the variables code a morphism. If we put the $\bot$ value for every remaining source vertex variable, we get a SEPI from the source graph to the target graph. □

An enumeration of the source vertex variables is enough to enforce arc surjectivity. However, compared to enumerating the antecedents variables beforehand, the former strategy checks the surjectivity quite late. Enumerating antecedent variables is sufficient provided that we fill the remaining source vertex variables with $\bot$. This "antecedents first" strategy works best in practice.

6.5. Implementation and evaluation

The preceding constraint satisfaction algorithm can be directly coded in GNU-Prolog [7] using the built-in constraint propagators for the `relation` and `element_var` constraints. Some redundant constraints, such as the `all_different` constraint of [15], the `neighborhood` constraint of [11] or other global constraints [18,20,17] have been proposed to improve the domain filtering process on graph matching problems, and in some cases to outperform dedicated algorithms such as Vflib [20,17]. However, the results reported here concern the constraint logic program described in the previous section without global constraints.

Our benchmark for evaluation comes from the repository of models biomodels.net that is widely used in systems biology. Most of these models are biochemical reaction networks from which one can extract a bipartite reaction graph with 'species' and 'reaction' vertices. The domain constraints ensure that the morphism map species vertices to species vertices, and reaction vertices to reaction vertices.

We have shown in [8] that SEPIs faithfully represent reduction relationships between models of biochemical reaction systems and that they can be used to relate the models in this repository, and organize them in a hierarchy of more or less detailed models.

On the 241 curated models of this repository, 131 are reaction models from which non-trivial reaction graphs can be extracted. The average size of the graphs is 56 vertices, with a minimum of 9, a median of 37, and a maximum of 315 vertices.

Our GNU-Prolog program was used to decide the existence of SEPIs in all 131*130 pairs of reaction graphs. Of these 17030 comparisons, 329 were not computed within a time out of 20 mn, and 16659 were computed in less than 5 s [8].

7. Generalization to undirected graphs and bipartite graphs

7.1. Undirected graphs

In undirected graphs, with loops allowed, the definition of SEPI is almost the same, with epimorphisms now preserving adjacency instead of successors.

Proposition 3 shows that the classical encoding of undirected graphs as symmetric directed graphs helps the conversion to undirected graphs.

The notion of outgoing and incoming non-neighbors can be traded for a notion of non-neighbors. Proposition 4 can be translated for non-neighbors. The infinite antichain given as proof of Proposition 5 is symmetric, so it also translates as an infinite antichain of undirected graphs.

The operations we consider in the proofs for distances d_r preserve arc symmetry, which makes the lub characterization of distance work for undirected graphs.

The SEPI existence problem between two undirected graphs is also NP-complete since the proof of Theorem 25 uses symmetric graphs.

7.2. Bipartite graphs

Our interest in subgraph epimorphisms came from an application in bioinformatics that uses bipartite directed graphs. In this setting, epimorphisms have to preserve bipartition of the vertices. Here, every vertex has one of two labels (say 'circle' and 'square'), vertices with the same label cannot be adjacent, and epimorphisms have to preserve these labels.

The infinite antichain in Proposition 5 can be adapted: instead of taking complete graphs minus maximal cycles, we can consider complete (n, n)-bipartite graphs minus maximal cycles, that is: $G_n = (V_n, A_n)$, with $V_n = \{1 \dots n\} \uplus \{1' \dots n'\}$ and $A_n = \{(i, j'), (j', i) \mid i \neq j'\}$.

The distances are defined the same way. The lub characterizations are the same for d_d and d_m. However, the construction to go from EPI to SEPI needs a slight modification: we add two $\perp$ vertices instead of one, a circle-labeled one and a square-labeled one. The $\perp$ vertex of each type is to be linked with every vertex of the other type. This makes the encoding of SEPI in EPI valid.

Finally, the SEPI decision problem can be proved NP-complete by modifying the proof a little. First, we can suppose that there are no clauses that contains both x and $\neg x$ in the SAT instance. We take the same C and L as in the main proof; there are n clauses in the instance.

Then, we modify the source graph $\mathcal{G}_1 = (V_1, A_1)$ by changing cliques to bicliques:

$$\begin{aligned}
&\mathit{circle}(V_1) = L\\
&\mathit{square}(V_1) = C \cup B \quad \text{with } B = \{(i_1, j_1), (i_2, j_2) \in L^2 \mid \ell_{i_1,j_1} = \neg \ell_{i_2,j_2}\}\\
&A_1 = \{(c, l) \in C \times L \mid l = (c, \cdot)\} \cup \{(l, p) \in L \times B \mid p = (l, \cdot) \vee p = (\cdot, l)\}
\end{aligned}$$

the target graph pattern is then $\mathcal{G}_2 = (V_2, A_2)$

$$\begin{aligned}
&\mathit{circle}(V_2) = X' = \{x'_1, \dots, x'_n\}\\
&\mathit{square}(V_2) = C' \cup D', \quad \text{with } C' = \{c'_1, \dots, c'_n\} \text{ and } D' = \{d'_1, \dots, d'_n\}\\
&A_2 = \{(c'_i, x'_i) \mid 1 \leq i \leq n\} \cup \{(x'_i, d'_i) \mid 1 \leq i \leq n\}.
\end{aligned}$$

If the SAT instance is satisfiable, deducing a SEPI from $\mathcal{G}_1$ to $\mathcal{G}_2$ is trivial. If there is a SEPI μ from $\mathcal{G}_1$ to $\mathcal{G}_2$, then $\mu^{-1}(X')$ satisfies Lemma 24, concluding the proof.

8. Conclusion

The operations of deleting and merging vertices are natural operations for reducing a graph. While graph reductions through a sequence of vertex deletions (respectively mergings) characterize subgraph isomorphisms (respectively graph epimorphisms), sequences of both vertex deletion and merging operations characterize subgraph epimorphisms. Our proposal is thus to use subgraph epimorphism for comparing graphs in applications where a more flexible notion than the classical notion of subgraph isomorphism is required.

We have shown that SEPIs preserve graph completeness and arc symmetry and that, just like SISO and EPI, SEPI is not a well quasi-order. We have defined the SEPI, EPI and SISO distances between two graphs as the size of the largest SEPI (respectively EPI, SISO) lower bound graphs. These distances are equal to the minimum number of respectively vertex deletion and/or merging operations that are necessary to obtain isomorphic graphs. They are also metrics on graphs, and we have $d_d \geq d_{md}$ and $d_m \geq d_{md}$.

From a computational point of view, we have shown that the existence of a SEPI between two graphs is an NP-complete problem and have presented a constraint logic program for solving it with good performance in practice on a large benchmark of SEPI model reduction problems from systems biology.

It is worth noticing that, given two graphs G and G', the greatest lower SEPI bounds and the least upper SEPI bounds are also interesting to compute since they represent "intersection" and "union" graphs for the SEPI relation. For instance, in our motivating application in systems biology, these objects correspond to the intersection (respectively union) of models at different levels of details for a given biochemical process. These graphs are not unique but we are confident that the constraint satisfaction algorithm we have presented can be interestingly generalized to compute them.

Acknowledgments

We acknowledge support from the national OSEO BioIntelligence and ANR Calamar (ANR-08-SYSC-003) projects and fruitful discussions with the many participants of these projects.

References

[1] R. Ambauen, Stefan Fischer, Horst Bunke, Graph edit distance with node splitting and merging, in: IAPR Workshop on Graph-based Representation in Pattern Recognition, in: Lecture Notes in Computer Science, vol. 2726, Springer-Verlag, 2003, pp. 95–106.

[2] Béla Bollobás, Andrew G. Thomason, Hereditary and monotone properties of graphs, in: R.L. Graham, J. Nesetril (Eds.), The Mathematics of Paul Erdos, II, in: Algorithms and Combinatorics, vol. 14, Springer-Verlag, Berlin, 1997, pp. 70–78 (Chapter).

[3] Mieczyslaw Borowiecki, Peter Mihók, Hereditary properties of graphs, in: V.R. Kulli (Ed.), Advances in Graph Theory, Vishwa International Publications, Gulbarga, 1991, pp. 42–69.
[4] Horst Bunke, On a relation between graph edit distance and maximum common subgraph, Pattern Recognition Letters 18 (8) (1997) 689–694.
[5] Pierre-Antoine Champin, Christine Solnon, Measuring the similarity of labeled graphs, in: ICCBR 2003, in: LNAI, vol. 2689, Springer, 2003, pp. 80–95.
[6] Stephen A. Cook, The complexity of theorem-proving procedures, in: STOC, 1971, pp. 151–158.
[7] Daniel Diaz, GNU Prolog user's manual, 1999–2003.
[8] Steven Gay, Sylvain Soliman, François Fages, A graphical method for reducing and relating models in systems biology, Bioinformatics 26 (18) (2010) i575–i581. Special issue ECCB'10.
[9] Pavol Hell, Jaroslav Nesetril, Graphs and Homomorphisms, Oxford University Press, 2004.
[10] Ralf Hofestädt, A petri net application to model metabolic processes, Systems Analysis Modelling Simulation 16 (1994) 113–122.
[11] Javier Larrosa, Gabriel Valiente, Constraint satisfaction algorithms for graph pattern matching, Mathematical Structures in Computer Science 12 (4) (2002) 403–422.
[12] Vianney Le Clément, Yves Deville, Christine Solnon, Constraint-based graph matching, in: 15th International Conference on Principles and Practice of Constraint Programming, CP 2009, in: Lecture Notes in Computer Science, vol. 5732, Springer-Verlag, Lisbon, Portugal, 2009, pp. 274–288.
[13] László Lovász, Graph minor theory, Bulletin of the American Mathematical Society 43 (1) (2006) 75–86.
[14] Venkatramana N. Reddy, Michael L. Mavrovouniotis, Michael N. Liebman, Petri net representations in metabolic pathways, in: Lawrence Hunter, David B. Searls, Jude W. Shavlik (Eds.), Proceedings of the 1st International Conference on Intelligent Systems for Molecular Biology, ISMB, AAAI Press, 1993, pp. 328–336.
[15] Jean-Charles Régin, A filtering algorithm for constraints of difference in CSPs, in: AAAI 1994, 1994, pp. 362–367.
[16] Neil Robertson, Paul D. Seymour, Graph minors, XX. Wagner's conjecture, Journal of Combinatorial Theory, Series B 92 (2) (2004) 325–357.
[17] Christine Solnon, All different-based filtering for subgraph isomorphism, Artificial Intelligence 174 (12–13) (2010) 850–864.
[18] Sébastien Sorlin, Christine Solnon, A parametric filtering algorithm for the graph isomorphism problem, Constraints 13 (4) (2008) 518–537.
[19] Narayan Vikas, Computational complexity of graph compaction, Ph.D. Thesis, Simon Fraser University, August 1997.
[20] Stéphane Zampelli, Yves Deville, Christine Solnon, Solving subgraph isomorphism problems with constraint programming, Constraints 15 (3) (2010) 327–353.

3.3 A Constraint Solving Approach to Tropical Equilibration

For smaller models where the kinetics have to be taken into account, methods based on the underlying ODE system exist. Note that even for moderately small models, the question of reduction remains crucial in that it allows the modeller to identify with much more confidence the crucial kinetic parameters, prompting biological experiments. This is especially true if one wants to apply some mathematical biology methods like bifurcation analysis, which mostly brings information for models with dimension smaller than three.

In this article we describe the constraint-based implementation of an algorithm computing tropical equilibrations for a polynomial ODE system. The result is a piecewise reduced model, using different fast/slow decompositions of variables in different regions of the space. Though it is already clear that the different reduced models do correspond to SEPI reductions, a more complete link delineating conditions on the kinetics for SEPIs would be of interest. In any case, the hybrid models resulting from a tropical reduction remain in the realm of some of our analyses, since we have proven recently that they can be encoded in SBML or BIOCHAM [3].

[19] Sylvain Soliman, François Fages, and Ovidiu Radulescu. "A constraint solving approach to model reduction by tropical equilibration". In: *Algorithms for Molecular Biology* 9.24 (Dec. 2014). ISSN: 1748-7188. DOI: 10.1186/s13015-014-0024-2

Soliman *et al. Algorithms for Molecular Biology* 2014, **9**:24
http://www.almob.org/content/9/1/24

AMB ALGORITHMS FOR MOLECULAR BIOLOGY

RESEARCH Open Access

A constraint solving approach to model reduction by tropical equilibration

Sylvain Soliman[1*], François Fages[1] and Ovidiu Radulescu[2]

Abstract

Model reduction is a central topic in systems biology and dynamical systems theory, for reducing the complexity of detailed models, finding important parameters, and developing multi-scale models for instance. While singular perturbation theory is a standard mathematical tool to analyze the different time scales of a dynamical system and decompose the system accordingly, tropical methods provide a simple algebraic framework to perform these analyses systematically in polynomial systems. The crux of these methods is in the computation of tropical equilibrations. In this paper we show that constraint-based methods, using reified constraints for expressing the equilibration conditions, make it possible to numerically solve non-linear tropical equilibration problems, out of reach of standard computation methods. We illustrate this approach first with the detailed reduction of a simple biochemical mechanism, the Michaelis-Menten enzymatic reaction model, and second, with large-scale performance figures obtained on the http://biomodels.net repository.

Keywords: Systems biology, Model reduction, Tropical algebra, Tropical equilibration, Constraint programming

Background

Model reduction is a central topic in systems biology and dynamical systems theory, for reducing the complexity of detailed models, finding important parameters, and developing multi-scale models for instance.

Indeed, for many of the problems in computation and analysis of complex systems, the upper limit on the size of the system that can be studied has been reached. This limit can be very low, namely tens of variables for system identification, symbolic calculation or bifurcation of attractors of large dynamical systems. For instance, the complexity of extant symbolic solvers of polynomial equations is exponential in the number of indeterminates and parameters, that sets a drastic limitation to the size of the models that can be analyzed [1,2]. Some examples of computational difficulties that arise when trying to apply standard tools of algebraic geometry to systems biology models can be found in [3]. Model reduction is a way to bypass these limitations by replacing large scale models with models containing less parameters and variables, easier to analyse.

There are mathematical methods, based on singular perturbations or on the theory of invariant manifolds, allowing reduction of fully parametrized systems with separation of time scales. More precisely, in dissipative systems, fast variables relax rapidly to some low dimensional attractive manifold called invariant manifold [4] that carries the slow mode dynamics. A projection of dynamical equations onto this manifold provides the reduced dynamics. Numerical reduction methods, such as computational singular perturbation (CSP, [5]), intrinsic low dimensional manifold (ILDM, [6]) exploit the separation of timescales of various processes and compute approximations of the invariant manifold. Purely structural reduction methods can handle big models possibly with lack of kinetic information [7]. However, the case of biochemical models of intermediate size, with partially known parameters and that ask for symbolic analysis, is more open [8].

While singular perturbation theory is a standard mathematical tool to analyze the different time scales of a dynamical system and decompose the system accordingly, tropical methods provide a simple algebraic framework to perform these analyses systematically in polynomial systems, and in situations when model parameters are known only by their orders of magnitude. Differential

*Correspondence: Sylvain.Soliman@inria.fr
[1]Inria, Domaine de Voluceau 78150, Rocquencourt, France
Full list of author information is available at the end of the article

equations describing kinetics of biochemical reactions are polynomial or become polynomial after decomposition of reaction mechanisms into elementary steps. For these models, quasi-equilibrium or quasi-steady state invariant manifolds allowing reductions are given by systems of algebraic equations [3]. A potentially crucial application of tropical mathematics is to enumerate and describe asymptotic solutions of algebraic systems of equations [9]. In particular, tropical solutions of polynomial equations provide the leading terms of their solutions via curves or in other terms via Newton-Puiseux series [10,11]. At the basis of our method lies the idea that equilibration of fast variables on invariant manifolds implies, at lowest order, equilibration of at least two dominant monomials, one positive and the other negative in the right hand side of the differential equations. We have called such a condition, similar to Kapranov's condition [11] for existence of Newton-Puiseux series with specified lowest order terms, tropical equilibration. The crux of our method lies in the computation of tropical equilibrations that define some reduced truncated systems with fewer parameters to identify, thus pointing to fewer experiments to (in)validate the model [12,13]. Our method copes with uncertain parameters by replacing exact values by orders of magnitude and the reduction is performed symbolically in both variables and parameters. With respect to methods based on singular perturbations, this could be less precise at lowest order, but it is more general in implementation.

Solving the tropical equilibration problem boils down to solving a system of equations in the min-plus algebra (also known as the tropical semiring). For solving linear tropical systems there are pseudo-polynomial algorithms, i.e. whose complexity is polynomial in the size of the system and in the absolute values of its coefficients [14]. In the nonlinear case, the existence of tropical equilibrations, which is equivalent to the problem of the intersection of tropical varieties, was shown to be NP-complete [15]. In this paper we show that constraint-based methods, using reified constraints for expressing the equilibration conditions, make it possible to numerically solve non-linear tropical equilibration problems, out of reach of standard computation methods.

We first illustrate this approach with the detailed reduction of a simple biochemical mechanism, the Michaelis-Menten enzymatic reaction model. We detail the general procedure to obtain truncated systems by identification, through equilibration, of fast and slow species, and relate the obtained reduced systems to the usual notions of quasi-steady-state and quasi-equilibrium. Then, we demonstrate that the approach is computationally feasible, and scales up properly, by treating in an automatic way all the curated dynamical models of http://biomodels.net repository [16].

Model reduction by tropicalization

We consider networks of biochemical reactions with mass action kinetic laws. The structure of a reaction is defined by a multiset rewriting rule as

$$\sum_{i=1}^{n} \alpha_{ji} A_i \rightarrow \sum_{k=1}^{n} \beta_{jk} A_k$$

where A_i, $i = 1, \ldots, n$ denote the chemical species and α_{ji}, β_{jk} are positive integers named stoichiometric coefficients defining which species are consumed and produced by the reaction j, $1 \leq j \leq r$, and in which quantities.

The mass action law means that reaction rates are monomial functions of the species concentrations x_i, $1 \leq i \leq n$ and read

$$R_j(\boldsymbol{x}) = k_j \boldsymbol{x}^{\boldsymbol{\alpha}_j},$$

where $k_j > 0$ are kinetic parameters and we use the shorthand notation $\boldsymbol{x}^{\boldsymbol{\alpha}_j} = x_1^{\alpha_{j1}} \ldots x_n^{\alpha_{jn}}$.

The network dynamics is then described by the following differential equations

$$\frac{dx_i}{dt} = \sum_{j=1}^{r} k_j S_{ij} \boldsymbol{x}^{\boldsymbol{\alpha}_j}. \tag{1}$$

where $S_{ij} = \beta_{ji} - \alpha_{ji}$ are entries of the stoichiometric matrix.

In what follows, the kinetic parameters do not have to be known precisely, they are given by their orders of magnitude. A convenient way to represent orders is by considering that

$$k_j = \bar{k}_j \epsilon^{\gamma_j},$$

where ϵ is a positive parameter much smaller than 1, γ_j is an integer or, more generally, a rational number, and $\bar{k}_j$ has order unity. An approximate integer order can be obtained from any real positive parameter by

$$\gamma_j = \text{round}(\log(k_j)/\log(\epsilon)),$$

where round stands for the closest integer. Notice that in this representation, small quantities have large orders. Furthermore, the smaller ϵ, the better the separation between quantities of different orders, indeed $\lim_{\epsilon \to 0} \frac{k_i}{k_j} = \infty$, if $\gamma_i < \gamma_j$.

We also define orders for species concentrations, using a vector of orders $\boldsymbol{a} = (a_1, \ldots, a_n)$, such that $\boldsymbol{x} = \bar{\boldsymbol{x}} \epsilon^{\boldsymbol{a}}$. We suppose that various $(a_1, \ldots, a_n)$ are integers or rational numbers with a common denominator. In our method we will calculate the concentration orders as solutions of the tropical equilibration problem (see below).

First, let us replace Eqs. (1) by their equivalent rescaled versions

$$\frac{d\bar{x}_i}{dt} = \sum_{j=1}^{r} \epsilon^{\mu_j - a_i} k_j S_{ij} \bar{\boldsymbol{x}}^{\boldsymbol{\alpha}_j}, \tag{2}$$

Soliman *et al. Algorithms for Molecular Biology* 2014, **9**:24
http://www.almob.org/content/9/1/24

where

$$\mu_j = \gamma_j + < \boldsymbol{a}, \boldsymbol{\alpha_j} >, \tag{3}$$

and $<,>$ stands for the vector dot product.

The r.h.s. of each equation in (2) is a sum of multivariate monomials in the concentrations. The orders μ_j indicate how large are these monomials, in absolute value. For sufficiently small ϵ, monomials of different orders are well separated. For instance a monomial of smallest order $\mu_j < \mu_{j'}$ dominates the other monomials $k_j|S_{ij}|\boldsymbol{x}^{\boldsymbol{\alpha_j}} \gg k_{j'}|S_{ij'}|\boldsymbol{x}^{\boldsymbol{\alpha_{j'}}}$. One could see all these monomials as "forces" acting on the chemical species. At steady state, the resultant of all these forces should be naught. A consequence of this is that the orders of dominant positive and negative forces should be equal. This is exactly our notion of *tropical equilibration* that we introduced in [13]. More precisely, we say the system (2) is *tropically equilibrated* iff

$$\min(\mu_j, S_{ij} < 0) = \min(\mu_{j'}, S_{ij'} > 0), \text{ for all } \quad i = 1, \ldots, n \tag{4}$$

The tropical equilibration problem consists in solving the system (4) for orders a_i, $1 \leq i \leq n$.

Another way to understand the condition (4) is via Newton-Puiseux series. Suppose we want to solve the polynomial equation

$$P(x, \epsilon) = \sum_j b_j \epsilon^{\gamma_j} x^{\alpha_j} = 0, \tag{5}$$

where α_j are positive integers, γ_j are rational powers, and b_j are real coefficients. It is well known [10] that solutions of this equation can be expressed as Newton-Puiseux series, i.e. have the form

$$x(\epsilon) = c_1 \epsilon^{\frac{a_1}{q}} + c_2 \epsilon^{\frac{a_2}{q}} + \ldots,$$

where c_i are complex coefficients, $a_1 < a_2 < \ldots$ are integers, q is a positive integer. By substituting $x(\epsilon) = c_1 \epsilon^{\frac{a_1}{q}} (1 + x_1(\epsilon))$ (where $x_1(\epsilon)$ collects terms with positive orders in ϵ) in (5) we get

$$P(x, \epsilon) = \sum_j b_j c_1^{\alpha_j} \epsilon^{\gamma_j + a_1 \alpha_j / q} + r_1(\epsilon) = 0,$$

where $r_1(\epsilon)$ collects higher order terms. Necessary conditions for $P(x, \epsilon) = 0$ read at lowest order

$$\sum_{j, \gamma_j + a_1 \alpha_j / q = m} b_j c_1^{\alpha_j} = 0 \tag{6}$$

$$m = \min_j (\gamma_j + a_1 \alpha_j / q) \tag{7}$$

In order to satisfy (6), the minimum in (7) should be attained at least twice. Furthermore, if one looks for real solutions $c_i \in \mathbb{R}$, then from (6) it follows that at least two b_j corresponding to the minimum (7) should have opposite signs. This means that the lowest order a_1/q in the Newton-Puiseux series solution has to satisfy a tropical equilibration problem.

We must emphasize that the tropical equilibration condition is weaker than the steady state condition, and makes sense also away from a steady state. In systems with slow/fast variables, the fast variables are equilibrated by compensation of dominant forces whose orders result from the tropical condition (4). As a consequence, the fast variables can be expressed as functions of the slow variables. However, both fast and slow variables can slowly evolve under the influence of weaker, higher order forces. This picture is valid as long as the relative dominance relations between various monomial terms in Eqs.(2) are preserved. This is true if the rescaled concentrations $\bar{x}_i$ stay between bounds, whereas ϵ is allowed to tend to zero.

To summarize, the tropical equilibration is a necessary condition for the elimination of fast variables and model reduction. As we showed in [13], in order to become sufficient this condition should be combined with a boundedness condition, called permanency:

Definition 0.1. *The system* (1) *is permanent, if there are two constants* $C_- > 0$ *and* $C_+ > 0$*, a set of renormalization exponents* a_i*, and a function* T_0 *of the initial conditions, such that after renormalization we have*

$$C_- < \bar{x}_i(t) < C_+, \quad \textit{for all } t > T_0(\boldsymbol{x}(0)) \textit{ and for every } i.$$

A simple example, the Michaelis-Menten reduction

The Michaelis-Menten enzymatic reaction network consists of three reactions:

$$S + E \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} ES \overset{k_2}{\to} P + E,$$

where S, ES, E, P represent the substrate, the enzyme-substrate complex, the enzyme and the product, respectively.

The corresponding system of polynomial differential equations reads:

$$\begin{aligned} x_1' &= -k_1 x_1 x_3 + k_{-1} x_2 \\ x_2' &= k_1 x_1 x_3 - (k_{-1} + k_2) x_2 \\ x_3' &= -k_1 x_1 x_3 + (k_{-1} + k_2) x_2 \\ x_4' &= k_2 x_2 \end{aligned} \tag{8}$$

where $x_1 = [S], x_2 = [ES], x_3 = [E], x_4 = [P]$.

It can be easily checked that the system has two algebraic invariants: $(x_2 + x_3)' = 0$, which implies

$$x_2 + x_3 = e_0, \tag{9}$$

where e_0 is a positive constant (the total amount of enzyme), and

$$x_1 + x_3 + x_4 = s_0 \tag{10}$$

where s_0 is a positive constant (the total amount of substrate and product). These conservation laws can be used to reduce the model by elimination of x_4 (by (10)) and x_3 (by (9)). It follows:

$$\begin{aligned} x_1' &= -k_1 x_1(e_0 - x_2) + k_{-1}x_2 \\ x_2' &= k_1 x_1(e_0 - x_2) - (k_{-1} + k_2)x_2 \end{aligned} \tag{11}$$

The constraint $x_2 \leq e_0$ resulting from the elimination is also to consider, but we will see that all equilibrations of the above equations already imply it.

Tropical equilibration equations

By rescaling variables and parameters, we get $x_i = \bar{x}_i \epsilon^{a_i}$, $1 \leq i \leq 2$, $k_1 = \bar{k}_1 \epsilon^{\gamma_1}$, $k_{-1} = \bar{k}_{-1}\epsilon^{\gamma_{-1}}$, $e_0 = \bar{e}_0 \epsilon^{\gamma_e}$.

The tropical equilibration equations for the reduced system read:

$$\gamma_1 + \gamma_e + a_1 = \min(\gamma_1 + a_1, \gamma_{-1}) + a_2 \tag{12}$$

$$\gamma_1 + \gamma_e + a_1 = \min(\gamma_1 + a_1, \min(\gamma_{-1}, \gamma_2)) + a_2 \tag{13}$$

The set of integer (or rational) orders endowed with the $\oplus = \min$ and $\otimes = +$ operations is a semi-field, called min-plus algebra or tropical semi-field [17]. In this semi-field, $-\infty$ plays the role of 0 and 0 plays the role of 1. The multiplicative inverse of a is denoted $-a$. Our tropical equilibration problem means solving a set of polynomial equations in this semi-field. Using these notations and properties of semi-field operation, the tropical equations become:

$$\gamma_1 \otimes \gamma_e \otimes a_1 \otimes (-a_2) = (\gamma_1 \otimes a_1) \oplus \gamma_{-1} \tag{14}$$

$$\gamma_1 \otimes \gamma_e \otimes a_1 \otimes (-a_2) = (\gamma_1 \otimes a_1) \oplus \gamma_{-1} \oplus \gamma_2 \tag{15}$$

Classical Michaelis-Menten reduction

The classical derivation of the Michaelis-Menten reduction is based on the behaviour of the variable x_2 for the complex concentration. Using (8) it follows that:

$$x_4' = V_m x_2 / e_0$$

where $V_m = k_2 e_0$ is the maximum value of the production rate x_4', since $x_2 \leq e_0$.

The variable x_2 satisfies equilibration relations and can be expressed as a function of a slow variable (either the substrate x_1 when x_2 is small, or the sum $x_1 + x_2$ in general) in two situations: quasi-stationarity and quasi-equilibrium.

The quasi-stationarity corresponds to setting x_2' to zero and is justified by the smallness of x_2 that can be considered a fast species (radical). More precisely one has $k_1 x_1(e_0 - x_2) - (k_{-1} + k_2)x_2 = 0$, leading to $x_2 = x_1 e_0/(K_m + x_1)$, where $K_m = (k_{-1} + k_2)/k_1$, i.e.

$$x_4' = V_m x_1/(K_m + x_1) \tag{16}$$

The quasi-equilibrium corresponds to setting $k_1 x_1 (e_0 - x_2) - k_{-1}x_2 = 0$, meaning zero net flux of the first reaction in the mechanism. This leads to $x_2 = x_1 e_0/((k_{-1}/k_1) + x_1)$, i.e.

$$x_4' = V_m x_1/((k_{-1}/k_1) + x_1) \tag{17}$$

This is justified by having a very fast transformations in the direct and reverse sense by the first reaction, much faster than the transformations by the second reaction. In this case both x_1 and x_2 are fast, but their sum $x_1 + x_2$ is slow.

We show next, in Section "Tropical equilibrations and model reductions", that analysis of tropical equations provide the conditions for the asymptotic validity of quasi-stationarity and quasi-equilibrium approximations and also the exhaustive list of asymptotic regimes.

Geometrical interpretation

It was discussed in [13] that there is a bijection between the set of solutions of each tropical equation and parts of the tropical curves of the polynomials defining the ordinary differential equations. A tropical curve is defined as the locus of species concentration values (x, y) where at least two monomials of the considered polynomial are equal and larger than all the others. In logarithmic scale, this locus is made of lines, half-lines, or line segments [13,18]. There is one tropical curve for each differential equation. For instance, the tropical curve defined by the polynomial $-k_1 e_0 x_1 + k_1 x_1 x_2 + k_{-1} x_2$ is made of three half-lines with a common origin depicted in Figure 1, namely

$$\log(x_2) = \log(e_0), \qquad \log(x_1) > \log(k_{-1}/k_1) \tag{18}$$

$$\log(x_1) = \log(k_{-1}/k_1), \qquad \log(x_2) > \log(e_0) \tag{19}$$

$$\log(x_2) = \log(x_1) + \log(e_0 k_1/k_{-1}), \quad \log(x_1) < \log(k_{-1}/k_1) \tag{20}$$

The solutions of the tropical equation (14) form two branches, corresponding to the two situations $(\gamma_1 \otimes a_1) \oplus \gamma_{-1} = (\gamma_1 \otimes a_1)$ and $(\gamma_1 \otimes a_1) \oplus \gamma_{-1} = \gamma_{-1}$, respectively. These are two half-lines in the plane of concentration orders:

$$a_2 = \gamma_e, \qquad \gamma_1 + a_1 < \gamma_{-1} \tag{21}$$

$$a_2 = a_1 + \gamma_1 + \gamma_e - \gamma_{-1}, \qquad \gamma_1 + a_1 > \gamma_{-1} \tag{22}$$

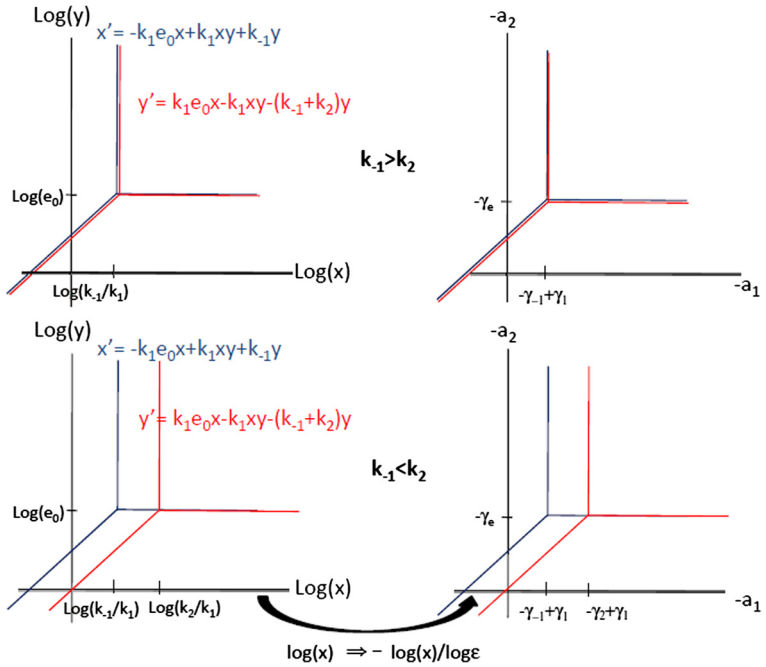


Figure 1 Tropical curves in the planes of concentrations and orders for the two variables Michaelis-Menten model. Tropical curves are defined as the locus of points where two monomials of a polynomial describing a differential equation are equal. The tropical curves for each differential equations are indicated by colors, blue for the first equation and red for the second equation. The vertical half-line of each of the tripods does not carry tropical equilibrations because it corresponds to equality of two monomials of the same sign. The horizontal and the oblique half-lines of the tripods carry tropical equilibrations. We have represented the two situations when $k_{-1} > k_2$ and when $k_{-1} < k_2$. All the tropical equilibrations are double (both variables are equilibrated) in the first case, and can be simple (only one variable is equilibrated) in the latter.

The two branches of solutions can be also related to parts of the tropical curves corresponding to the equilibration of monomials of different signs. More precisely (21) corresponds to (18), and (22) corresponds to (20). The branch (19) of the tropical curve corresponds to the equality of two positive monomials and has no correspondence in the set of tropical equilibrations.

Similarly to computing steady states as intersections of null-clines, we are considering multiple tropical equilibrations as intersections of tropical curves.

We therefore consider the second tropical equation (15), in two situations $\gamma_{-1} \oplus \gamma_2 = \gamma_{-1}$ and $\gamma_{-1} \oplus \gamma_2 = \gamma_2$. In the first case the tropical equation (15) is equivalent to the tropical equation (14) (also, the tropical curves coincide). Therefore, the two solutions (21) and (22) equilibrate both equations. In the second case, the solutions of the tropical equation (15) form two branches, corresponding to $(\gamma_1 \otimes a_1) \oplus \gamma_2 = \gamma_1 \otimes a_1$ and $(\gamma_1 \otimes a_1) \oplus \gamma_2 = \gamma_2$, respectively. They correspond to two half-lines in the plane of orders (a_1, a_2), namely $a_2 = \gamma_e, \quad a_1 < \gamma_2 - \gamma_1$ and $a_2 = a_1 + \gamma_1 + \gamma_e - \gamma_2, \quad a_1 > \gamma_2 - \gamma_1$. A simple graphical inspection of the relative positions of these half-lines with respect to the half-lines carrying solutions of the first tropical equation shows that there are four branches of tropical equilibrations:

$$a_2 = \gamma_e, \qquad a_1 < \gamma_2 - \gamma_1 \tag{23}$$

$$a_2 = \gamma_e, \qquad \gamma_2 - \gamma_1 < a_1 < \gamma_2 - \gamma_1 \tag{24}$$

$$a_2 = a_1 + \gamma_1 + \gamma_e - \gamma_2, \qquad a_1 > \gamma_2 - \gamma_1 \tag{25}$$

$$a_2 = a_1 + \gamma_1 + \gamma_e - \gamma_{-1}, \qquad a_1 > \gamma_{-1} - \gamma_1 \tag{26}$$

The branch (23) equilibrates the two variables. The branch (25) equilibrates only the second variable, whereas the branches (24), (26) equilibrate only the first variable.

Tropical equilibrations and conservation laws

The reduced Michaelis-Menten mechanism with two dynamical variables has been obtained by elimination of a variable using an exact conservation law. It is interesting to compute the tropical equilibrations directly, in the unreduced model. In this three variables model, two of the equilibrium equations are identical. Like for computation of steady states, we need the conservation law as an extra constraint. If we treat this constraint exactly, we obtain the reduced model. An approximate treatment of Eqs. (8), (9),

considering equilibration of dominant terms, leads to the tropical problem:

$$\gamma_1 \otimes a_1 \otimes a_3 = \gamma_{-1} \otimes a_2 \tag{27}$$

$$\gamma_1 \otimes a_1 \otimes a_3 = (\gamma_{-1} \otimes a_2) \oplus (\gamma_2 \otimes a_2) \tag{28}$$

$$a_2 \oplus a_3 = \gamma_e \tag{29}$$

This tropical problem is different from (14), (15) and leads to different solutions in general. Firstly, let us notice that elimination is not possible in semi-fields, because there is no additive inverse in general. Hence, (27), (28) (29) can not be reduced to an equivalent system of two tropical equations. Secondly, dominant monomial equilibration in the reduced model does not always correspond to monomial equilibrations in the unreduced model. A typical example is the monomial x_1x_3 that becomes the difference $x_1e_0 - x_1x_2$ in the reduced model. The two monomials can equilibrate each other in the reduced model, but the same quantity is an unique, unequilibrated monomial in the full model.

There are six branches of tropical solutions of the system (27), (28), (29). Two branches are obtained when $\gamma_{-1} \otimes a_2 = \gamma_{-1}$. In this case the two tropical equations (27), (28) are identical. Depending on $a_2 \oplus a_3 = a_2$, or $a_2 \oplus a_3 = a_3$ we get:

$$a_2 = \gamma_e, \qquad a_1 < \gamma_{-1} - \gamma_1 \tag{30}$$

$$a_2 = a_1 + \gamma_1 + \gamma_e - \gamma_{-1}, \qquad a_1 > \gamma_{-1} - \gamma_1 \tag{31}$$

These branches correspond to equilibrations of all the variables.

When $\gamma_{-1} \otimes a_2 = \gamma_2$ the two tropical Eqs. 27, (28) are incompatible. Depending on $a_2 \oplus a_3 = a_2$, or $a_2 \oplus a_3 = a_3$ and further choosing only one of the two tropical Eqs. 27, (28) we get the following branches:

$$a_2 = \gamma_e, \qquad a_1 < \gamma_{-1} - \gamma_1 \tag{32}$$

$$a_2 = \gamma_e, \qquad a_1 < \gamma_2 - \gamma_1 \tag{33}$$

$$a_2 = a_1 + \gamma_1 + \gamma_e - \gamma_{-1}, \qquad a_1 > \gamma_{-1} - \gamma_1, \tag{34}$$

$$a_2 = a_1 + \gamma_1 + \gamma_e - \gamma_2, \qquad a_1 > \gamma_2 - \gamma_1 \tag{35}$$

In the branches (32), (34), the variables x_2, x_3 are not equilibrated, whereas in the branches (33), (35), the variable x_1 is not equilibrated.

Comparison of Eqs. (30)-(35) and Eqs. (21)-(26) proves that the tropical equations of the unreduced model have the same set of solutions as the reduced model. However, the branch of solutions (33) equilibrates all the variables in the reduced model and does not equilibrate the variable x_1 in the reduced model. The reason is exactly the one given above: the monomial x_1x_3 is dominant and unequilibrated in the unreduced model, becomes $x_1e_0 - x_1x_2$ with equilibrated monomials in the reduced model.

Tropical equilibrations and model reductions

Tropical equilibrations can be used for model reduction. The reduction starts by tropical truncation. We call tropically truncated model the model obtained by elimination of all dominated monomials from the r.h.s. of the ordinary differential equations. The next step is ordering the variables according to the values of the exponents $\mu_i - a_i$. This allows to determine which variables are slow and fast.

An additional construction is needed in the case when the tropically truncated system of fast variables has conservation laws that are not conserved by the un-truncated system. The conservation laws define species pools that are supplementary slow variables. The pools follow differential equations involving previously dominated monomials.

For instance, in the two variables Michaelis-Menten model, we found essentially two types of reduced models, corresponding to quasi-equilibrium and quasi-stationarity approximations [19].

The branch (21) of tropical solutions leads to the following truncated system:

$$\begin{aligned} x_1' &= -k_1x_1e_0 + k_{-1}x_2 \\ x_2' &= k_1x_1e_0 - k_{-1}x_2 \end{aligned} \tag{36}$$

This truncated system has conserved quantity $z = x_1 + x_2$. The variable z is not conserved by the full model described by (11). Indeed, addition of Eqs. (11) leads to $z' = -k_{-1}x_2$. As the variable x_2 can be eliminated from $-k_1x_1e_0 + k_{-1}x_2 = 0$ and $x_1 + x_2 = z$ we have the reduced dynamics $z' = -k_zz$, where $k_z = k_{-1}/(1 + k_{-1}/(k_1e_0))$. For small ϵ, we can consider that $k_z \sim \epsilon^{\gamma_z}$, with $\gamma_z = \gamma_{-1} - \min(0, \gamma_{-1} - \gamma_1 - \gamma_e)$. Because $\mu_1 - a_1 = \gamma_1 + \gamma_e$, $\mu_2 - a_2 = \gamma_1 + \gamma_e + a_1 - a_2 = \gamma_{-1}$ the relation $k_z > \mu_1 - a_1$, $k_z > \mu_2 - a_2$ are always satisfied guaranteeing that z is slower than x_1, x_2. The form (36) of the truncated equations and the conservation of $x_1 + x_2$ by the fast dynamics shows that this case corresponds to quasi-equilibrium of the first reaction in the Michaelis-Menten model, as described in Section "Classical Michaelis-Menten reduction", equation 17.

The branches (23) and (24) lead to quasi-equilibrium with the following truncated system:

$$\begin{aligned} x_1' &= -k_1x_1(e_0 - x_2) \\ x_2' &= k_1x_1(e_0 - x_2) \end{aligned} \tag{37}$$

These cases correspond to saturation of the enzyme (x_2 has the same order as e_0). A slow variable $z = x_1 + x_2$ has to be introduced as before, but the reduced dynamics is $z' = -k_{-1}x_2 = -k_{-1}e_0$.

The branch (25) leads to quasi-stationarity of the enzyme/substrate complex. In this case we have the tropical truncated system:

$$\begin{aligned} x_1' &= -k_1 x_1 e_0 \\ x_2' &= k_1 x_1 e_0 - k_2 x_2 \end{aligned} \tag{38}$$

The variable x_1 is not equilibrated, which still allows for model reduction because this variable is slow. The fast variable x_2 is equilibrated, and the equilibration equation corresponds to the classical notion of quasi-stationary approximation, as described in Section "Classical Michaelis-Menten reduction", equation 16. In this case, $\mu_1 - a_1 = \gamma_1 + \gamma_e$, $\mu_2 - a_2 = \gamma_1 + \gamma_e + a_1 - a_2 = \gamma_2$. The condition that x_1 is slower than x_2 reads $\mu_1 - a_1 > \mu_2 - a_2$ and we get the additional condition $\gamma_1 + \gamma_e > \gamma_2$, which is a low enzyme concentration condition.

The branch (26) leads to quasi-stationarity of the substrate with the following truncated system:

$$\begin{aligned} x_1' &= -k_1 x_1 e_0 + k_{-1} x_2, \\ x_2' &= -k_2 x_2 \end{aligned} \tag{39}$$

The variable x_2 is not equilibrated, which is allowed only if this variable is slower than x_1. In this case, $\mu_1 - a_1 = \gamma_1 + \gamma_e$, $\mu_2 - a_2 = \gamma_2$. The condition that x_2 is slower than x_1 reads $\mu_1 - a_1 < \mu_2 - a_2$ and leads to the additional condition $\gamma_1 + \gamma_e < \gamma_2$, which is a high enzyme concentration condition.

Finally, the branch (24) leads to the truncated system:

$$\begin{aligned} x_1' &= -k_1 x_1 (e_0 - x_2) \\ x_2' &= -k_2 x_2 \end{aligned} \tag{40}$$

The variable x_1 is equilibrated, but it can not satisfy permanency. Indeed, at fixed x_2 this variable will converge to zero. Therefore, this tropical equilibration does not lead to a reduced model.

Tropical equilibration as a constraint satisfaction problem

As explained in Section "Model reduction by tropicalization", given a biochemical reaction system with its Mass-Action kinetics, and a small ϵ, the problem of tropical equilibration is to look for a rescaling of the variables such that the dominating positive and negative term in each ODE *equilibrate* as per Eqs. (4), i.e., are of the same degree in ϵ.

Section "A simple example, the Michaelis-Menten reduction" showed that it is possible to iteratively reduce the equilibration problem to a linear system of equations for each possible pair of positive and negative dominating monomial. It is actually possible to consider fewer pairs by restricting that search to the pairs denoting edges of the Newton polygon [13]. Nevertheless, the number of linear systems to consider remains exponential in the number of species, and may lead to redhibitory computational costs, especially when handling biochemical systems with hub molecules, i.e., molecules involved in a high number of reactions (e.g., E2F, p53, cMyc in cell-cycle control or NFκB in signalling), which corresponds to a Newton polygon with many vertices.

In order to tackle that complexity, we propose a numerical approach based on Constraint Programming, that encodes the equilibration problem as a Constraint Satisfaction Problem (CSP) [20-22] and uses reified constraints to prune the search space. Constraint Programming is a paradigm that has showed great success at solving combinatorial decision or optimization problems, in particular for real-world instances of NP-hard problems, e.g., in the field of production planning and scheduling. It is therefore an interesting way to approach the combinatorial explosion described above.

In presence of invariants (conservation laws) in the original system, Section "Conservation law constraints" has shown that some constraints related to rescaling need be added. We have shown in [23] that finding these conservation laws can be efficiently solved by constraint methods. Here we will thus assume that the conservation laws are given in input. In our prototype implementation, both the computation of conservation laws and the following equilibration are performed for a given system.

Reified constraints

Key to the modeling of tropical equilibration problems as CSP are reified constraints. Reified constraints are special constraints that link in a bidirectional way the value of a boolean variable to the satisfaction of another constraint. They allow for powerful cuts in the search space by propagating the truth value of some constraints of the problem to the truth value of the Boolean variable, and vice versa.

For instance, the reified constraint

B# <==> X# = Y

states that the Boolean variable B is true (i.e. equal to 1) if and only if the constraints $X = Y$ is satisfied. That constraint posts the constraint $X = Y$ (resp. $X \neq Y$) as soon as B gets value 1 (resp. 0), and vice versa, sets $B = 1$ (resp. $B = 0$) as soon as $X = Y$ (resp. $X \neq Y$ i.e. when the domains of X and Y become disjoint).

For the tropical equilibration problem, these constraints are at the core of our representation of the minimum constraints as they enforce the propagation of existing knowledge before branching on the two possible values. Indeed, if A is the minimum of B and C, you can derive

many things on the domains of A, B and C before eventually trying $A = B$ or $A = C$. For instance it is safe to add that $A \leq B$ and $A \leq C$, but also if you have, from other equations, that $B \geq min_B$ and $C \geq min_C$ then you can add the fact that $A \geq min(min_B, min_C)$. If later you obtain that actually $A = B$ then you can enforce $C \geq B$, etc. Section "Minimum constraints" shows in more detail how reified constraint do precisely this kind of conditional addition of cuts and can therefore be used to handle minimum constraints while postponing enumerative search as much as possible.

Variables and domains

For practical reasons, mainly the lack of an efficient solver over rationals with reified constraints, we use a finite domain solver and therefore only look for integer solutions (whereas solutions are rational). In practice this did not seem to change much the nature of results, see Figure 2. Extensions of the approach to cope with half-integer solutions or with rational solutions with a common, small denominator are straightforward.

For each original equation dx_i/dt, $1 \leq i \leq n$ is introduced a variable $a_i \in \mathbb{Z}$ that is used to rescale the system by posing $x_i = \epsilon^{a_i}\bar{x_i}$. These are the variables of our CSP. Note that they require a solver handling $\mathbb{Z}$ like for instance SWI-Prolog [24,25] with the `clpfd` library by Markus Triska, which we used for our implementation.

Tropical equilibration constraints

For each differential equation that should be equilibrated is a list of positive monomials M_i^+, and a list of negative monomials M_i^-. The degrees in ϵ of all these monomials are integer linear expressions in the a_i. Now, to obtain an equilibration one should enforce for each i that the minimum degree in M_i^+ is equal to the minimum degree in M_i^-. This corresponds to the Eqs. 4. We will see how they can be implemented with reified constraints, but for now, let us assume a constraint min(L, M)| that enforces that the variable M of $\mathbb{Z}$ is the minimum value of a list L of linear expressions over variables of $\mathbb{Z}$. We have in our CSP, for each $1 \leq i \leq n$, min(PositiveMonomialDegrees, M) and min(NegativeMonomialDegrees, M).

Conservation law constraints

The second kind of constraint comes from conservation laws. Each conservation law is an equality between a linear combination of the x_i and a constant c_i. By rescaling, we obtain a sum of rescaled monomials equal to $\epsilon^{\log(c_i)/\log(\epsilon)}\bar{c_i}$. We want this equality to hold when ϵ goes to zero, which implies that the minimal degree in ϵ in the left hand side is equal to (the round of) the degree of the right hand side. Since once again the degrees on the left are linear combinations of our variables a_i, this is again a constraint of the form: min(ConservationLawDegrees, K) where K is equal to round($\log(c_i)/\log(\epsilon)$). This corresponds to the tropical equation (29).

Minimum constraints

Furthermore, if the system under study is not at steady state, the minimum degree should not be reached only once, which would lead to a constant value for the corresponding variable when ϵ goes to zero, but at least twice. This is the case for the example treated in [12]. The constraint we need is therefore slightly more general than min/2: we need the constraint min(L, M, N) which is true if M is smaller than each element of L and equal to N elements of that list. Note that using CLP notation, we have:

$$\text{min(M, L)} \quad :- \quad \text{C\# >= 1}, \quad \text{min(M, L, C)}.$$

In order to enforce that the minimum is reached at least a required number of times, one obvious solution is to try all pairs of positive and negative monomials and

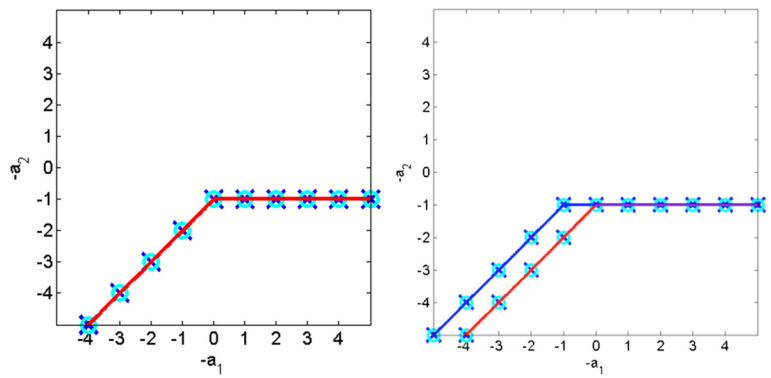


Figure 2 Comparison of the theoretical and computed equilibrations in the cases $k_{-1} > k_2$ and $k_{-1} < k_2$. The circles are equilibrations computed for the simplified two variables Michaelis-Menten model, the crosses are for the full three variables model. The lines indicate the theoretical equilibrations.

count the successful pairs [26]. However, this is not necessary, the min(L, M, N) constraint directly expresses the cardinality constraint on the minimums and can be implemented using *reified constraints* to propagate information between L, M and N in all directions, without enumeration. Using SWI-Prolog notations, the implementation of min/3 by reified constraints is as follows:

```
min([ ] ,, 0).
min([ HT] , M, C) : −M# =< H, B# <==>
  M# = H, C# = B + CC, min(T, M, CC).
```

The translation of this predicate into words is roughly as follows, first ignoring the counts: M is smaller than a list with head H and tail T, if it is smaller than the tail T and it is smaller than the head, i.e., M≤H. Now, we also impose that the value M is reached C times as follows: it is reached CC times in the tail and C = B + CC where B is a variable equal to 1 iff M is equal to the head and 0 otherwise. Note that this latest statement is enforced through a reified constraint, it will therefore not lead to immediate branching but to the propagation of as much information as possible (e.g., if some values in the list are already known to be strictly greater than others, the corresponding boolean for each of them will be set to 0, and thus the sum will by necessity enforce some other values to be equal to the required minimum).

This concise and portable implementation will probably improve when the minimum and $\min_n$ global constraints are available (see [27] for a reference). However it already proves very efficient as demonstrated in the next section.

When C is equal to one, we can fall back to using the built-in min construct in a constraint (e.g., M #= min(L1, min(L2, L3))). Some preliminary benchmarking showed that the reified version is more efficient if the length of the list is greater than 3 or 4.

Enumeration strategy

Constraints over finite domains come with domain filtering algorithms which dynamically prune the domain of variables when the domain of other variables change in a constraint. However this strategy is not complete and must be combined with a search procedure for virtually enumerating all possible values of the variables. For this application we obtained good performances with dichotomic search by bissecting the domain of the variables (bisect option in SWI-Prolog) without any particular heuristics for choosing the variables.

Note that since this approach is numerical, contrary to solving *symbolically* an exponential but finite number of linear systems as done in Section "A simple example, the Michaelis-Menten reduction" and in [13], there can be an infinite number of solutions. This situation denotes an under-constrained linear system and remains to be interpreted biologically. In practice bounds are put on variables in order to force finiteness. This is not a restriction in practice since biochemical species' concentrations usually do not vary by more than a hundred of magnitude orders.

Furthermore, in order to speed-up the computation of all solutions in such large domains, we used an iterative domain expansion strategy: the problem is first tried with a domain of $[-2, 2]$ for all variables , i.e., equilibrations are searched by rescaling in the $10^{-2}, 10^{2}$ interval. If that fails, the domain is doubled and the problem tried again until a limit of $10^{-128}, 10^{128}$.

Computation results on Biomodels.net

To benchmark our approach, we applied it systematically to all the dynamical models of the curated part of the http://biomodels.net repository [16] of biological systems, with ϵ set arbitrarily to 0.1.

We used release *r24* from 2012-12-12 which includes 436 curated models. Among them, only 55 models have non-trivial purely polynomial kinetics (ignoring *events* if any). Our computational results on those are summarized in Table 1, where the first column indicates whether a complete equilibration was found, and the times are in seconds.

The domain expansion strategy coupled with dichotomic search by domain bisections allowed us to gain two orders of magnitude of computation time on the biggest models.

Only one of the models (number 002) used values far from 0 in the equilibration (up to ϵ^{40}) and has no complete equilibration if the domain is restricted to $[-32, 32]$. This is because the model is written with units such that the initial concentrations are of the order 10^{-21}, translating the search accordingly. We thus do not believe that enlarging the domains even more would lead to more equilibrations. Nevertheless, choosing a smaller ϵ might increase the number of equilibrations.

18 of the 23 models for which there is a complete equilibration are actually under-constrained and appear to have an infinity of such solutions (typically linear relations between variables). For the 5 remaining ones, we computed all complete equilibrations as shown in Table 2.

Table 1 Number of models of the BioModels repository, with a polynomial kinetics, for which tropical equilibrations were found or not, with corresponding size of the model and computation time

Found	# models	Variables (avg/min/max)	Time in seconds (avg/min/max)
yes	23	17.348/3/ 86	0.486/0.004/2.803
no	32	17.812/1/194	0.099/0.000/1.934

Table 2 Number of equilibrations and computation time for the models of the BioModels database with finitely many numerical solutions

Model	# variables	# equilibrations	Total time (s)
BIOMD0000000002	13	18	53
BIOMD0000000122	14	9	4.1
BIOMD0000000156	3	1	<1ms
BIOMD0000000229	7	1	0.07
BIOMD0000000413	5	5	0.4

Conclusions

In this paper we have shown that constraint-based methods can be efficiently used to numerically solve tropical equilibration problems in biological models of real-size in the BioModels.net repository. These calulations are important for model reduction and for determining the unknown orders of the variables. Once the orders of the variables are known, the rapid variables can be identified and the system reduced to a simpler one. This truncation, described in Section "Tropical equilibrations and model reductions" coupled with the proposed constraint-based method for finding equilibrations therefore provides an automatic way to reduce models and to identify fast/slow variables. We have started the application of such technique on non-trivial models provided by biologists and modellers and hope to be able to improve both the understanding, through that identification of fast/slow variables, and the analysis, through the size reduction, of those models.

Even with the progress of high-throughput technologies, having more focused models, with fewer species and parameters to measure, will definitely permit an improvement in the quality and speed of development of the models. Furthermore, the structural methods for comparing models in model repositories, such as [7], can be refined by filtering the structural reduction relationships according to the kinetics of the reactions and the tropical reasoning on the magnitude orders.

In many cases, it makes sense biologically to only look for partial equilibrations. Strategies to decide when such decision has to be made remain unclear. Nevertheless the framework of partial constraint satisfaction and more specifically Max-CSP [28] would allow us to easily handle the maximization of the number of equilibrated variables.

One of the limits of this approach, is that it is not particularly well suited to equilibration problems with an infinite number of solutions. As discussed at the end of previous section, in such situations symbolic solutions would be more appropriate. Nevertheless, even the approximate detection of such a case by the very high number of (bounded) numerical solutions was shown to be not very costly in practice.

Competing interests
The authors declare that they have no competing interests.

Authors' contributions
FF and OR designed the study. SS devised the algorithm and conducted the experiments. All authors equally contributed to the writing of the manuscript. All authors read and approved the final manuscript.

Acknowledgements
This work has been supported by the French ANR BioTempo, CNRS Peps ModRedBio, EPIGENMED Excellence Laboratory and OSEO Biointelligence projects.

Author details
[1]Inria, Domaine de Voluceau 78150, Rocquencourt, France. [2]University of Montpellier 2, Place Eugene Bataillon, 34095 Montpellier, France.

Received: 4 April 2014 Accepted: 4 November 2014
Published online: 04 December 2014

References

1. Grigoriev D, Vorobjov N: **Solving systems of polynomial inequalities in subexponential time.** *J Symbolic Computat* 1988, **5:**37–64.
2. Grigoriev D: **Complexity of quantifier elimination in the theory of ordinary differential equations.** *Lect Notes Comput Sci* 1989, **18:**11–25.
3. Pantea C, Gupta A, Rawlings JB, Craciun G: **The QSSA in chemical kinetics: as taught and as practiced.** In *Discrete and Topological Models in Molecular Biology*. Berlin: Springer; 2014:419–442.
4. Gorban A, Karlin I: **Invariant manifolds for physical and chemical kinetics.** *Lect Notes Phys* 2005, **660:**1–491.
5. Lam S, Goussis D: **The CSP method for simplifying kinetics.** *Int J Chem Kinet* 1994, **26**(4):461–486.
6. Maas U, Pope SB: **Simplifying chemical kinetics: intrinsic low-dimensional manifolds in composition space.** *Combustion Flame* 1992, **88**(3):239–264.
7. Gay S, Soliman S, Fages F: **A graphical method for reducing and relating models in systems biology.** *Bioinformatics* 2010, **26**(18):i575–i581. [Special issue ECCB'10].
8. Radulescu O, Gorban AN, Zinovyev A, Noel V: **Reduction of dynamical biochemical reactions networks in computational biology.** *Front Genet* 2012, **3:**131. [http://www.frontiersin.org/bioinformatics_and_computational_biology/10.3389/fgene.2012.00131/abstract]
9. Sturmfels B: *Solving systems of polynomial equations, Volume 97*. American Mathematical Soc: Providence; 2002.
10. Walker RJ: *Algebraic curves*. New York: Springer; 1978.
11. Einsiedler M, Kapranov M, Lind D: **Non-archimedean amoebas and tropical varieties.** *J für die reine und angewandte Mathematik (Crelles J)* 2006, **2006**(601):139–157.
12. Noel V, Grigoriev D, Vakulenko S, Radulescu O: **Tropical geometries and dynamics of biochemical networks application to hybrid cell cycle models.** *Electron Notes Theor Comput Sci* 2012, **284:**75–91.
13. Noel V, Grigoriev D, Vakulenko S, Radulescu O: **Tropicalization and tropical equilibration of chemical reactions.** In *Tropical and Idempotent Mathematics and Applications, Volume 616 of Contemporary Mathematics*. Edited by Litvinov G, Sergeev S: American Mathematical Society; 2014:261–277.
14. Grigoriev D: **Complexity of solving tropical linear systems.** *Comput Complexity* 2013, **22:**71–88.
15. Theobald T: **On the frontiers of polynomial computations in tropical geometry.** *J Symbolic Comput* 2006, **41**(12):1360–1375.
16. le Novère N, Bornstein B, Broicher A, Courtot M, Donizelli M, Dharuri H, Li L, Sauro H, Schilstra M, Shapiro B, Snoep JL, Hucka M: **BioModels Database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems.** *Nucleic Acid Res* 2006, **1**(34):D689–D691.
17. Cohen G, Gaubert S, Quadrat J: **Max-plus algebra and system theory: where we are and where to go now.** *Ann Rev Control* 1999, **23:**207–219.
18. Viro O: **From the sixteenth Hilbert problem to tropical geometry.** *Jpn J Math* 2008, **3**(2):185–214.

19. Gorban AN, Radulescu O, Zinovyev AY: **Asymptotology of chemical reaction networks.** *Chem Eng Sci* 2010, **65**(7):2310–2324. [International Symposium on Mathematics in Chemical Kinetics and Engineering].
20. Mackworth AK: **Consistency in networks of relations.** *Artif Intell* 1977, **8:**99–118.
21. Meseguer P: **Constraint satisfaction problems: an overview.** *A.I. Commun* 1989, **2:**3–17.
22. Kumar V: **Algorithms for constraint- satisfaction problems: a survey.** *A.I. Mag* 1992, **13:**32–44.
23. Soliman S: **Invariants and other structural properties of biochemical models as a constraint satisfaction problem.** *Algorithms Mol Biol* 2012, **7**(15):15.
24. Wielemaker J, Schrijvers T, Triska M, Lager T: **SWI-Prolog.** *Theory Prac Logic Program* 2012, **12**(1-2):67–96.
25. Wielemaker J: *SWI-Prolog 6.3.15 Reference Manual*; 1990. [http://www.swi-prolog.org/pldoc/refman/]
26. Radulescu O, Gorban A, Zinovyev A, Noel V: **Reduction of dynamical biochemical reaction networks in computational biology.** *Front Bioinformatics Comput Biol* 2012, **3:**131.
27. Beldiceanu N, Carlsson M, Demassey S, Petit T: **Global constraints catalog.** Tech. Rep. T2005-6, Swedish Institute of Computer Science 2005.
28. Freuder EC, Wallace RJ: **Partial constraint satisfaction.** *Artif Intell* 1992, **58:**21–70.

doi:10.1186/s13015-014-0024-2
Cite this article as: Soliman *et al.*: A constraint solving approach to model reduction by tropical equilibration. *Algorithms for Molecular Biology* 2014 **9**:24.

L'homme de génie est celui qui m'en donne.

Paul Valéry

Chapter 4

One Structure, a Hierarchy of Semantics

Summary

The two previous chapters have been going back and forth between different semantics of biochemical models, from purely graphical to quantitative and dynamical. The fact that the very same model can be interpreted in different ways is at the core of software like BIOCHAM. It is also a fact spread all along the SBML specification. For instance, about kinetic laws it reads:

> In this section, we provide an interpretation of SBML kinetic laws in the framework of a system of ordinary differential equations (ODEs). However [...] it is equally possible to translate a model into other frameworks, and some formulations, such as discrete stochastic systems, are indeed quite common. (SBML Level 3 Core specification, Section 4.11.7 – Mathematical interpretation of SBML reactions and kinetic laws)

In this chapter we present formal attempts that link the different semantics of biochemical systems. This takes the form of a hierarchy of semantics for reaction models, but also of inference of influences/regulations from those same reactions, as used in our multistationarity analysis in Section 2.4. Using the framework of Abstract Interpretation [28] we define abstractions, but the case of concretizations is also interesting, especially when trying to find a *reasonable* model for a given system of ODE.

This chapter is composed of four papers:

[6] François Fages and Sylvain Soliman. "Abstract Interpretation and Types for Systems Biology". In: *Theoretical Computer Science* 403.1 (2008), pp. 52–70. DOI: 10.1016/j.tcs.2008.04.024 : a paper on abstract interpretation in systems biology for defining a hierarchy of semantics for reaction systems, and relating them to influence models, published in *Theoretical Computer Science*, 403(1), 2008.

[7] François Fages and Sylvain Soliman. "From reaction models to influence graphs and back: a theorem". In: *Proceedings of Formal Methods in Systems Biology FMSB'08*. Lecture Notes in Computer Science 5054. Springer-Verlag, Feb. 2008. DOI: 10.1007/978-3-540-68413-8_7 : a more readable version of the relationship between reaction and influence models, published in FMSB'08, Springer Verlag.

[5] François Fages, Steven Gay, and Sylvain Soliman. "Inferring Reaction Systems from Ordinary Differential Equations". In: *Theoretical Computer Science* 599 (Sept. 2015), pp. 64–78. ISSN: 0304-3975. DOI: 10.1016/j.tcs.2014.07.032 : a paper on the inference of reaction systems from ordinary differential equations accepted in *Theoretical Computer Science*, 2014, special issue of CMSB 2012 (extended version of [4])

[20] Sylvain Soliman and Monika Heiner. "A Unique Transformation from Ordinary Differential Equations to Reaction Networks". In: *PLoS One* 5.12 (Dec. 2010), e14284. DOI: 10.1371/journal.pone.0014284 : unicity conditions for the inference of reaction systems from ODEs, published in PLoS One, 2010.

The first two ones focus on the hierarchy of semantics, on the abstractions and type inference properties, whereas the last two ones focus on the inference of biochemical reaction models for a system of ODEs.

4.1 Abstraction and Type Inference

4.1.1 Abstract Interpretation and Types for Systems Biology

Abstract Interpretation provides a formal mathematical framework to relate the different possible semantics of a biochemical model represented, e.g., in SBML or BIOCHAM. This article shows how the boolean, discrete, stochastic and continuous semantics of a single model are related and then builds on that formal foundation to propose some abstractions seen as types on such models. The neighborhood type is to be checked against the *outside* annotation of SBML compartments, whereas the more usual protein functions (kinase/phosphatase) or influences are also described as types.

Note that BIOCHAM implements even more type checking, since dimensional analysis[1] is also a type checking/inference task [45] that can actually be stated as a CSP.

[6] François Fages and Sylvain Soliman. "Abstract Interpretation and Types for Systems Biology". In: *Theoretical Computer Science* 403.1 (2008), pp. 52–70. DOI: 10.1016/j.tcs.2008.04.024

[1] http://lifeware.inria.fr/biocham/DOC/manual.html#tth_sEc3.8.2

Theoretical Computer Science 403 (2008) 52–70

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Abstract interpretation and types for systems biology☆

François Fages, Sylvain Soliman*

Project-team Contraintes, INRIA Paris-Rocquencourt, BP105, 78153 Le Chesnay Cedex, France

ARTICLE INFO

Article history:
Received 3 August 2007
Received in revised form 31 January 2008
Accepted 20 April 2008
Communicated by G. Rozenberg

Keywords:
SBML
Type inference
Hierarchy of semantics
Ordinary differential equations
Markov chains

ABSTRACT

Abstract interpretation is a theory of abstraction that has been introduced for the analysis of programs. In particular, it has proved useful for organizing the multiple semantics of a given programming language in a hierarchy corresponding to different detail levels, and for defining type systems for programming languages and program analyzers in software engineering. In this paper, we investigate the application of these concepts to systems biology formalisms. More specifically, we consider the Systems Biology Markup Language SBML, and the Biochemical Abstract Machine BIOCHAM with its differential, stochastic, discrete and boolean semantics. We first show how all of these different semantics, except the differential one, can be formally related by simple Galois connections. Then we define three type systems: one for checking or inferring the functions of proteins in a reaction model, one for checking or inferring the activation and inhibition effects of proteins in a reaction model, and another one for checking or inferring the topology of compartments or locations. We show that the framework of abstract interpretation elegantly applies to the formalization of these further abstractions, and to the implementation of linear or quadratic time type checking as well as type inference algorithms. Furthermore, we show a theorem of independence of the graph of activation and inhibition effects from the kinetic expressions in the reaction model, under general conditions. Through some examples, we show that the analysis of biochemical models by type inference provides accurate and useful information. Interestingly, such a mathematical formalization of the abstractions commonly used in systems biology already provides some guidelines for the extensions of biochemical reaction rule languages.

1. Introduction

Systems biology aims at elucidating the high-level functions of the cell from their biochemical basis at the molecular level [27]. A lot of work has been done for collecting genomic and post-genomic data and making them available in databases [1,28], and for organizing the knowledge on pathways and interaction networks into models of cell metabolism, signaling, cycle, apoptosis, etc. now published in model repositories (e.g. http://biomodels.net/). Furthermore the Systems Biology Markup Language (SBML) [26] provides a common exchange format for reaction models, which is nowadays supported by the majority of modeling tools [25,39].

Models of biological processes are built with two somewhat contradictory perspectives that are worth clarifying. The first perspective is one of knowledge representation. In this perspective, the more concrete the better: models aim at gathering, in a consistent way, the current knowledge on particular systems and at representing the interactions participating in a system

☆ This article is an extended version of [François Fages, Sylvain Soliman, Type inference in systems biology, in: Corrado Priami (Ed.), CMSB'06: Proceedings of the Fourth International Conference on Computational Methods in Systems Biology, in: Lecture Notes in Computer Science, vol. 4210, Springer-Verlag, 2006].

* Corresponding author. Tel.: +33 1 39635761; fax: +33 1 39635469.
E-mail addresses: Francois.Fages@inria.fr (F. Fages), Sylvain.Soliman@inria.fr (S. Soliman).

0304-3975/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2008.04.024

with the maximum of details. The second perspective for building models is to make predictions and answer particular questions about a system. Yet in this perspective, the more abstract the better: models for making predictions should get rid of useless details and should represent the minimum information that is sufficient for answering the questions; the less the information the more powerful and efficient the tools available.

One way to reconcile these two perspectives is to put more focus on the issue of abstraction in systems biology, and to develop not only models but also their relationships to other models at different abstraction levels. In this paper we propose a formal ground for this issue by transposing the concepts of abstract interpretation and types borrowed from programming theory to systems biology. Abstract interpretation is a theory of abstraction, introduced by Cousot and Cousot in [15] as a framework for reasoning about programs, their semantics [14], and for designing static analyzers, among which type inference systems [13]. Type checking and type inference are important concepts and methods in programming languages and software engineering [5]. Type checking is a way to ensure some level of consistency, depending on the type system, in large programs and in complex assemblies of software components. Type inference provides powerful static analysis of pre-existing programs without types; it also facilitates the use of type systems by freeing the user from entering type information.

In this paper, we investigate the application of these concepts to systems biology formalisms. More specifically, we consider the Systems Biology Markup Language SBML [26] and the Biochemical Abstract Machine BIOCHAM [4,19] with its differential, stochastic, discrete and boolean semantics [3,17]. We first show how these different semantics can be formally related by simple Galois connections, as required in the theory of abstract interpretation, with the noticeable exception of the differential semantics that is discussed with some details.

Then we study three type systems:

(1) one for checking or inferring the protein functions in a reaction model,
(2) one for checking or inferring the activation and inhibition effects in a reaction model,
(3) and another one for checking or inferring the topology of compartments or locations in reaction models with space considerations.

We show that the framework of abstract interpretation elegantly applies to the formalization of these type abstractions, and to the implementation of linear or quadratic time complexity type checking as well as type inference algorithms. Furthermore, when comparing the inference of the activation and inhibition effects from the syntax of the reaction rules with their inference from the differential semantics, we show a theorem of independence of the graph of activation and inhibition effects from the kinetic expressions, under general conditions.

Through some examples of reaction models coming from the BioModels and BIOCHAM repositories [39], we show that the analysis of biochemical models by type inference provides accurate and useful information. Interestingly, we show that such a mathematical formalization of abstractions commonly used in systems biology already provides some guidelines for the extensions of biochemical reaction rule languages.

2. Preliminaries on abstract interpretation, type checking and type inference

2.1. Domains, abstractions and galois connections

In the algebraic setting of abstract interpretation, a domain is a lattice $L(\sqsubseteq, \bot, \top, \sqcup, \sqcap)$ defined by a partial order $(L, \sqsubseteq)$, where $\bot$ and $\top$, elements of L and $\sqcup, \sqcap$, binary operators on L, respectively denote the least element, the greatest element, the least upper bound and the greatest lower bound. Intuitively, the partial ordering represents the information loss: the lesser the more informative, the greater the bigger loss of information.

As it is often the case in program analysis, the concrete domain and the abstract domains considered for analyzing biochemical models, will be power-sets, i.e. set lattices $\mathcal{P}(\mathcal{S})(\subseteq, \emptyset, \mathcal{S}, \cup, \cap)$ ordered by inclusion, with the empty set as $\bot$ element, and the base set $\mathcal{S}$ as $\top$ element. For instance, in the syntactical domain of reaction rule sets ordered by inclusion, the base set of all possible reactions makes all behaviors possible and thus contains no information, while the empty set is the most precise in this information ordering.

An *abstraction* is formalized by a Galois connection between a concrete domain $\mathcal{C}$ and an abstract domain $\mathcal{A}$, as follows [15]:

Definition 1. A Galois connection $\mathcal{C} \underset{\gamma}{\overset{\alpha}{\longleftrightarrow}} \mathcal{A}$ between two lattices $(\mathcal{C}, \sqsubseteq_C)$ and $(\mathcal{A}, \sqsubseteq_\mathcal{A})$ is defined by an abstraction function $\alpha : \mathcal{C} \to \mathcal{A}$, and a concretization function $\gamma : \mathcal{A} \to \mathcal{C}$, that are monotonic:

- $\forall\, c, c' \in \mathcal{C} : c \sqsubseteq_\mathcal{C} c' \;\Rightarrow\; \alpha(c) \sqsubseteq_\mathcal{A} \alpha(c')$,
- $\forall\, a, a' \in \mathcal{A} : a \sqsubseteq_\mathcal{A} a' \;\Rightarrow\; \gamma(a) \sqsubseteq_\mathcal{C} \gamma(a')$,

and are adjoint:

- $\forall c \in \mathcal{C}, \forall a \in \mathcal{A} : c \sqsubseteq_\mathcal{C} \gamma(a) \Leftrightarrow \alpha(c) \sqsubseteq_\mathcal{A} a$.

For any Galois connection, we have the following properties:

(1) $\gamma \circ \alpha$ is extensive (i.e. $c \sqsubseteq_{\mathcal{C}} \gamma \circ \alpha(c)$) and represents the information lost by the abstraction
(2) $\alpha \circ \gamma$ is contracting (i.e. $\alpha \circ \gamma(a) \sqsubseteq_{\mathcal{A}} a$)
(3) $\gamma \circ \alpha$ is the identity *iff* γ is onto *iff* α is one-to-one
(4) α preserves $\sqcup$, and γ preserves $\sqcap$
(5) $\gamma(a) = \max \alpha^{-1}(\downarrow a) = \sqcup \alpha^{-1}(\downarrow a)$
(6) $\alpha(c) = \min \gamma^{-1}(\uparrow c) = \sqcap \gamma^{-1}(\uparrow c)$
(7) the composition of two Galois connections is a Galois connection.

where $\downarrow a = \{b \mid b \sqsubseteq a\}$ and $\uparrow a = \{b \mid a \sqsubseteq b\}$.

If $\gamma \circ \alpha$ is the identity, the abstraction α loses no information, and $\mathcal{C}$ and $\mathcal{A}$ are isomorphic from the information standpoint (although α may be not onto and γ not one-to-one). It is equivalent in the definition of Galois connections to replace the condition of adjointness by conditions 1 and 2, or by condition 5 which also entails the monotonicity of γ.

Furthermore, we shall use the fact that in powerset domains, the pointwise extension of any function from the base set of the concrete domain to the abstract domain forms a Galois connection:

Lemma 2. *Let $\mathcal{C}$ and $\mathcal{A}$ be two sets, and $\alpha : \mathcal{P}(\mathcal{C}) \longrightarrow \mathcal{P}(\mathcal{A})$ be a function such that $\alpha(c) = \bigcup_{e \in c} \alpha(\{e\})$. Then the function $\gamma(a) = \cup \alpha^{-1}(\downarrow a)$ forms a Galois connection $\mathcal{P}(\mathcal{C}) \underset{\gamma}{\overset{\alpha}{\longleftrightarrow}} \mathcal{P}(\mathcal{A})$ between $(\mathcal{P}(\mathcal{C}), \subseteq)$ and $(\mathcal{P}(\mathcal{A}), \subseteq)$.*

Proof. We show that α is monotonic and $\gamma(a) = \max \alpha^{-1}(\downarrow a)$.

The monotonicity of α is immediate since if $c \subseteq c'$ we have $\bigcup_{c_i \in c} \alpha(\{c_i\}) \subseteq \bigcup_{c_i \in c'} \alpha(\{c_i\})$.

Now, let us consider $c = \gamma(a) = \cup \alpha^{-1}(\downarrow a)$, we need to prove that $c \in \alpha^{-1}(\downarrow a)$, i.e. $\alpha(c) \in \downarrow a$. We know that $\alpha(c) = \bigcup_{e \in c} \alpha(\{e\}) = \bigcup_{e \in \cup \alpha^{-1}(\downarrow a)} \alpha(\{e\})$. For each e in $\cup \alpha^{-1}(\downarrow a)$ there exists $d \in \mathcal{P}(\mathcal{C})$ such that $e \in d$ and $\alpha(d) \subseteq a$, therefore $\alpha(\{e\}) \subseteq a$. Hence $\bigcup_{e \in \cup \alpha^{-1}(\downarrow a)} \alpha(\{e\}) \subseteq a$ and thus $\alpha(c) \subseteq a$. □

In this paper, we will consider the syntactical domain of reaction models ordered by the inclusion of rule sets as concrete domain, and four semantical domains for respectively:

- the stochastic semantics, in which the reaction rules are interpreted by a continuous time Markov chain;
- the discrete semantics, in which the rules are interpreted by a Petri net;
- the boolean semantics, in which the rules are interpreted by a boolean asynchronous transition system;
- and the differential semantics, in which the rules are interpreted by a system of ordinary differential equations.

We will show in Section 3 that, with the noticeable exception of the differential semantics, all these domains are formally related by simple Galois connections.

2.2. Type checking and type inference by abstract interpretation

Types provide further abstractions for reasoning about programs. In the setting of abstract interpretation, a type system $\mathcal{A}$ for a concrete domain $\mathcal{C}$ is nothing but a Galois connection $\mathcal{C} \underset{\gamma}{\overset{\alpha}{\longleftrightarrow}} \mathcal{A}$. The *type inference* problem is, given a concrete element $x \in \mathcal{C}$ (e.g. a reaction model), to compute $\alpha(x)$ (e.g. the protein functions that can be inferred from the reactions). The *type checking* problem is, given a concrete element $x \in \mathcal{C}$ and a typing $y \in \mathcal{A}$ (e.g. a set of protein functions), to determine whether $x \sqsubseteq_{\mathcal{C}} \gamma(y)$ (i.e. whether the reactions are compatible with the information given on the protein functions) which is equivalent to $\alpha(x) \sqsubseteq_{\mathcal{A}} y$ (i.e. whether the given typing contains the inferred types).

Most of the type systems considered in this paper will be implemented with type checking and type inference algorithms that basically browse the set of reactions, and check or collect the type information for each rule or pair of rules independently, thus in linear time or quadratic time respectively.

In this paper, we will consider three abstract domains for types:

- one for protein functions, where molecules are abstracted into categories such as kinases and phosphatases (Section 4),
- one for the influence graph, where the biochemical reaction rules are abstracted by binary relations of activation and inhibition between molecular species (Section 5),
- and one for location topologies, where reaction and transport rules are abstracted by retaining only the neighborhood information between locations (Section 6).

These domains will be defined by abstractions from the syntactical domain of reaction models. The syntactical domain indeed suffices to define the abstractions necessary for these analyses. It is worth noting that a similar situation also occurs in program analysis when the syntax of programs captures enough of the semantics for the needs of the analysis. For the analysis of influences between species, we will compare in Section 5 the results obtained by abstraction from the syntactical domain, with the information obtained by abstraction from the differential semantics.

3. Domains for reaction models and hierarchy of semantics

3.1. Syntactical domain of reaction models

Following SBML and BIOCHAM conventions, a model of a biochemical system is a set of reaction rules of the form e `for` l `=>` r where l is a multiset of molecule names given with stoichiometric coefficients, called a *solution*, r is the transformed solution, and e is a *kinetic expression*, i.e. a *positive* arithmetic expression on the concentrations of the molecules in i (plus possibly of some other molecules that have for instance an inhibitory effect on the reaction).

We will use the BIOCHAM operators `+` and `*` to denote solutions as `2*A + B`, as well as the syntax of catalyzed reactions `e for 2*A+B =[C]=> D` as an abbreviation for `e for 2*A+B+C => C+D`. By abuse of notation, assuming a finite set of molecules $\mathcal{M}$, we shall also see a solution l as an $|\mathcal{M}|$-dimensional vector of integers, and will denote by $l(A)$ the stoichiometric coefficient of A in solution l.

Formally, the concrete domain of reaction models is the powerset of all possible reaction rules ordered by set inclusion:

Definition 3. Given a finite set $\mathcal{M}$ of molecule names, the universe of reactions is the set of rules

$$\mathcal{R} = \{e \texttt{ for } l \texttt{ => } r \mid e \text{ is a kinetic expression, and } l \text{ and } r \text{ are solutions of molecules in } \mathcal{M}\}.$$

The concrete domain $\mathcal{D}_{\mathcal{R}} = (\mathcal{P}(\mathcal{R}), \subseteq)$ of *reaction models* is the power-set of reaction rules ordered by inclusion.

Note that in this domain, the composition of two reaction models is naturally the union of the sets of reactions. A reaction appearing in two reaction sets is thus not duplicated when composing two models by set union.

In the SBML exchange format, no particular semantics is defined, and this syntactical domain is the natural one to consider. In BIOCHAM, reaction models are interpreted under four semantics that correspond to four different abstraction levels: the boolean semantics, the discrete semantics, the differential semantics and the stochastic semantics [3,17]. In the following subsections, we formalize these semantical domains and study their formal relationship by Galois connections within a hierarchy of semantics.

It is worth noting that in the context of programming languages, it is not usual (and generally not possible) to include the syntactical domain ordered y set inclusion within the hierarchy of semantics. It is possible here however for the rule-based language of reactions, and should be possible as well for other rule-based languages in which programs can be ordered by set inclusion, like Prolog for instance [12].

3.2. Stochastic semantics

The most realistic interpretation of biochemical reaction models is provided by the *stochastic semantics*. In that semantics, a reaction model is interpreted as a (continuous time) Markov chain, and the kinetic expressions as transition rates. This interpretation is correct w.r.t. the Master Chemical Equation if we suppose that the reactions happen in a well stirred environment (i.e. "instantaneous" diffusion) with constant pressure, temperature and volume [24].

For a given volume V_k of the location where the compound x_k resides, a concentration C_k for x_k is translated into a molecule number $N_k = \lfloor C_k \times V_k \times N_A \rfloor$, where N_A is Avogadro's number. A state in the stochastic semantics will be a vector of integers indicating the numbers of molecules for each species.

Formally, given a fixed finite set $\mathcal{M}$ of molecule names, the stochastic transition semantics is defined by the following domain:

Definition 4. Let a *discrete state* be a vector of positive integers of dimension $|\mathcal{M}|$. The universe $\mathcal{S}$ of *stochastic transitions* is the set of triplets (S, S', τ) where S and S' are discrete states and $\tau \in \mathbb{R}^+$ is a weight. The domain $\mathcal{D}_{\mathcal{S}} = (\mathcal{P}(\mathcal{S}), \subseteq)$ of *stochastic transition models* is the power-set of stochastic transitions ordered by inclusion.

Note first that discrete states have the same mathematical structure as solutions in reaction rules, and can both be represented by $|\mathcal{M}|$-dimensional vectors of positive integers. In the following, we will identify states and solutions and will sum them (see definition of $S \rightarrow_i S'$ below and Theorem 13).

Note also that in a stochastic transition model s, there can be more than one transition from one state to another one, labelled with different real numbers. We define the *weight* in s of a transition from state S_i to S_j as the sum of the weights $\tau_{ij} = \sum_{(S_i,S_j,\tau)\in s} \tau$.

Now, an element s of the domain precisely defines a Markov chain where the *probability* p_{ij} of having a transition from state S_i to state S_j is obtained by normalizing the transition weights into $p_{ij} = \frac{\tau_{ij}}{\sum_k \tau_{ik}}$. Then the transition time can be computed as usual. Stochastic simulation techniques like Gillespie's algorithm [23] compute realizations of the processes described by models in the stochastic domain, where random variables range over the probability and the time of transition. The results of those simulations are generally noisy versions of the simulation obtained by the interpretation of the reaction rules by a system of ordinary differential equations (see Section 3.5). However, in models with for instance, very few molecules of some kind, qualitatively different behaviors may appear in the stochastic simulation, and thus justify the recourse to that semantics in such cases. A classical example is the model of the lambda phage virus [21] in which a small

number of molecules, promotion factors of two genes, can generate an explosive multiplication (lysis) after a more or less long period of passive wait (lysogeny).

Now, in order to relate the stochastic semantics domain to the syntactical domain of reaction rules, let us consider a reaction rule model $\{e_i \texttt{ for } l_i \texttt{ => } r_i\}_{i\in I}$, and denote by $S \to_i S'$ the fact that *rule i fires in state S* resulting in state S', i.e. if $S \geq l_i$ (pointwise) and $S' = S - l_i + r_i$.

In a given state S, the numbers of molecules are fixed integer values and the kinetic expression e_i evaluates into a (positive) real valued *reaction rate*, noted $e_i(S)$. This allows us to relate the stochastic transition domain to the syntactical domain of reaction rules by the following Galois connection:

Proposition 5. *Let $\alpha_{\mathcal{RS}} : \mathcal{D}_{\mathcal{R}} \to \mathcal{D}_{\mathcal{S}}$ be the function associating to a reaction model $\{e_i \texttt{ for } l_i\texttt{=>}r_i\}_{i\in I}$ the stochastic transition model $\{(S, S', e_i(S)) \in \mathcal{S} \mid i \in I,\ S \to_i S'\}$. Let $\gamma_{\mathcal{RS}}(s) = \cup{\alpha_{\mathcal{RS}}}^{-1}(\downarrow s)$. $\mathcal{D}_{\mathcal{R}} \underset{\gamma_{\mathcal{RS}}}{\overset{\alpha_{\mathcal{RS}}}{\longleftarrow\!\!\!\longrightarrow}} \mathcal{D}_{\mathcal{S}}$ is a Galois connection.*

Proof. Simply note that $\alpha_{\mathcal{RS}}$ is defined by its union on each rule of the concrete model and apply Lemma 2. □

Proposition 6. *$\alpha_{\mathcal{RS}}$ is not one-to-one.*

Proof. For instance, the reaction models $m1 = \{\texttt{ e for A => B}\}$ and $m2 = m1 \cup \{\texttt{ e for 2*A => A+B}\}$ have the same set of stochastic transitions. $\gamma \circ \alpha$ is thus not the identity, the information lost by the stochastic abstraction is the elimination of redundant rules in the reaction model. □

$\alpha_{\mathcal{RS}}$ is neither onto as the stochastic transitions obtained from a reaction model enjoy some particular properties, such as for instance the following stability property w.r.t. the number of molecules in the states:

Proposition 7. *If two states S_1, S_2 are such that $S_1 \leq S_2$ pointwise, then for any reaction model m and any stochastic transition $(S_1, S, \tau) \in \alpha_{\mathcal{RS}}(m)$, we have $(S_2, (S + S_2 - S_1), \tau) \in \alpha_{\mathcal{RS}}(m)$, i.e. all the rules that apply in S_1 apply in S_2 with the same effect.*

Proof. By definition of $\alpha_{\mathcal{RS}}$. □

Corollary 8. *$\alpha_{\mathcal{RS}}$ is not onto.*

3.3. Discrete semantics

The discrete semantics of reaction models can be defined as the trivial abstraction of the stochastic semantics that simply forgets the transition rates.

Definition 9. The universe $\mathcal{D}$ of *discrete transitions* is the set of pairs of discrete states. The domain $\mathcal{D}_{\mathcal{D}}$ of discrete transitions is the power-set of discrete transitions ordered by inclusion $\mathcal{D}_{\mathcal{D}} = (\mathcal{P}(\mathcal{D}), \subseteq)$.

Proposition 10. *Let $\alpha_{\mathcal{SD}} : \mathcal{D}_{\mathcal{S}} \to \mathcal{D}_{\mathcal{D}}$ be the function associating to a set of stochastic transitions the discrete transitions obtained by projection on the two first components, and $\gamma_{\mathcal{SD}}(d) = \cup{\alpha_{\mathcal{SD}}}^{-1}(\downarrow d)$. $\mathcal{D}_{\mathcal{S}} \underset{\gamma_{\mathcal{SD}}}{\overset{\alpha_{\mathcal{SD}}}{\longleftarrow\!\!\!\longrightarrow}} \mathcal{D}_{\mathcal{D}}$ is a Galois connection.*

Proof. Here again it suffices to note that $\alpha_{\mathcal{SD}}$ is defined by its union on each single stochastic transition of the concrete model and to apply Lemma 2. □

Remark that $\alpha_{\mathcal{SD}}$ is this time onto, but obviously not one-to-one as the transition rates are simply forgotten.

It is worth noticing that the discrete semantics corresponds to the classical Petri net semantics of reaction models [35,36, 9,22]. As a consequence, classical Petri net analysis tools can be used for the analysis of reaction models at this abstraction level. For instance, the elementary mode analysis of metabolic networks [37] has been shown in [44] to be equivalent to the classical analysis of Petri nets by T-invariants. These analyses apply to the discrete semantics of reaction models in all generality.

3.4. Boolean semantics

The boolean semantics is purely qualitative, and provides somehow the most abstract semantics of reaction models. The boolean semantics forgets the kinetic expressions and interprets the rules as a (non-deterministic) asynchronous transition system but this time over boolean states representing the absence or presence of molecules. It can be applied to large models for which the kinetic data may be not available.

Definition 11. Let a *boolean state* be a vector of booleans of dimension $|\mathcal{M}|$ indicating the presence of each molecule in the state. The universe $\mathcal{B}$ of *boolean transitions* is the set of pairs of boolean states. The domain $\mathcal{D}_{\mathcal{B}}$ of boolean transitions is the power-set of boolean transitions ordered by inclusion $\mathcal{D}_{\mathcal{B}} = (\mathcal{P}(\mathcal{B}), \subseteq)$.

This semantical domain is related to the discrete transitions semantics domain by the zero/non-zero abstraction from the integers to the booleans, and its pointwise extension from discrete states to boolean states $\alpha_{\mathcal{NB}} : \mathbb{N}^{|\mathcal{M}|} \to \mathbb{B}^{|\mathcal{M}|}$.

Proposition 12. *Let $\alpha_{\mathcal{DB}} : \mathcal{D}_{\mathcal{D}} \to \mathcal{D}_{\mathcal{B}}$ be the function associating to a set of discrete transitions the set of boolean transitions obtained by applying $\alpha_{\mathcal{NB}}$ to the discrete states. Let $\gamma_{\mathcal{DB}}(b) = \cup{\alpha_{\mathcal{DB}}}^{-1}(\downarrow b)$. $\mathcal{D}_{\mathcal{D}} \underset{\gamma_{\mathcal{DB}}}{\overset{\alpha_{\mathcal{DB}}}{\longleftarrow\!\!\!\longrightarrow}} \mathcal{D}_{\mathcal{B}}$ is a Galois connection.*

Proof. As before, note that $\alpha_{\mathcal{DB}}$ is defined by its union on each transition of the concrete model and apply Lemma 2. □

In BIOCHAM, the boolean semantics of reaction models is computed by associating to each reaction rule a set of boolean transition rules that take into account the possible complete consumption or not of the reactants by the reaction [7]. For instance, a reaction rule like `A+B=>C+D` is interpreted by four boolean transition rules:

- $A \wedge B \longrightarrow A \wedge B \wedge C \wedge D$
- $A \wedge B \longrightarrow \neg A \wedge B \wedge C \wedge D$
- $A \wedge B \longrightarrow A \wedge \neg B \wedge C \wedge D$
- $A \wedge B \longrightarrow \neg A \wedge \neg B \wedge C \wedge D$

Given a reaction model R, let us denote by S_{BB} the set of boolean transitions obtained by applying these boolean transition rules to each state. The following theorem shows that the BIOCHAM boolean semantics of reaction models *over-approximates* the boolean semantics obtained from the quantitative semantics. The non-existence of a behaviour in the BIOCHAM boolean semantics thus entails its non-existence in the quantitative semantics of the rules whatever the kinetic expressions are.

Theorem 13. *For any reaction model R, $\alpha_{\mathcal{DB}}(\alpha_{\mathcal{SD}}(\alpha_{\mathcal{RS}}(R))) \subseteq S_{BB}$.*

Proof. Since all our abstractions are defined pointwise, it is enough to prove it for only one rule in R. Let us consider e `for` l `=>` r. By abuse of notation we will denote by l and r the discrete states corresponding to solutions of same name. We have $\alpha_{\mathcal{RS}}(R) = \{(S_i, S_j, e) | S_i \geq l,\ S_j = S_i - l + r\}$ and thus $\alpha_{\mathcal{SD}}(\alpha_{\mathcal{RS}}(R)) = \{(S_i, S_j) | S_i \geq l,\ S_j = S_i - l + r\}$, which leads to $\alpha_{\mathcal{DB}}(\alpha_{\mathcal{SD}}(\alpha_{\mathcal{RS}}(R))) = \{(S'_i, S'_j) | S_i \geq l,\ S_j = S_i - l + r,\ S'_i = \alpha_{\mathcal{NB}}(S_i),\ S'_j = \alpha_{\mathcal{NB}}(S_j)\}$. Since $S_{BB} = \{(T, T') | T \geq \alpha_{\mathcal{NB}}(l),\ \alpha_{\mathcal{NB}}(r) \vee (T \wedge \neg\alpha_{\mathcal{NB}}(l)) \leq T' \leq \alpha_{\mathcal{NB}}(T) \vee \alpha_{\mathcal{NB}}(r)\}$ we can see that the property holds as $S_i \geq l$ implies $S'_i \geq \alpha_{\mathcal{NB}}(l)$, and since $S_i \geq l$ we have $S_j = S_i - l + r \Rightarrow S_i - l + r \leq S_j \leq S_i + r \Rightarrow \alpha_{\mathcal{NB}}(S_i - l + r) = \alpha_{\mathcal{NB}}(r) \vee (\alpha_{\mathcal{NB}}(S_i) \wedge \neg\alpha_{\mathcal{NB}}(l)) \leq S'_j \leq \alpha_{\mathcal{NB}}(S_i + r) = \alpha_{\mathcal{NB}}(S_i) \vee \alpha_{\mathcal{NB}}(r)$. □

It is worth noticing that this property does not hold for the boolean semantics of reaction models that always assume either incomplete consumption, or complete consumption, like in Pathway Logic [16] or in boolean Petri nets [22]. In these formalisms, the correctness of the boolean interpretation w.r.t. a quantitative interpretation is thus left to the modeler who is in charge of explicitly adding reaction rules for the different cases of consumption of the reactants.

3.5. Differential semantics

The *differential semantics* of reaction models interprets a set of reaction rules $\{e_i$ `for` l_i `=>` $r_i\}_{i=1,\ldots,n}$ over molecular concentration variables $\{x_1, \ldots, x_m\}$, by the following system of Ordinary Differential Equations (ODE):

$$\mathrm{d}x_k/\mathrm{d}t = \sum_{i=1}^{n} r_i(x_k) * e_i - \sum_{j=1}^{n} l_j(x_k) * e_j$$

where $r_i(x_k)$ (resp. l_i) is the stoichiometric coefficient of x_k in the right (resp. left) member of rule i. Thanks to its wide range of mathematical tools, this semantics is the most commonly used in mathematical biology [38].

The study of the relationship between the differential and the stochastic semantics dates back to the seminal work of Boltzmann in the XIXth century who created the domain of statistical physics. In this setting, the differential semantics is obtained from the stochastic semantics by limit operations where the number of molecules tends to the infinity and the time steps tend to zero. under several assumptions such as perfect diffusion.

In the setting of abstract interpretation, the differential semantics is however difficult to formally relate to the previous semantics for several reasons. The differential semantics is a synchronous semantics in the sense that it specifies the evolution of variables in parallel, whereas all the other semantics are asynchronous in the sense that the interleaving semantics is considered where one reaction is fired at a time. Hence the notion of time is not the same in both categories of semantics, having infinitesimal time steps in the differential semantics, and time for one transition in the other semantics. Furthermore the differential semantics is deterministic and produces an average trace, whereas the other semantics produce sets of possible traces representing the competition between reactions.

For these reasons, the differential semantics does not belong to our hierarchy of syntactical, stochastic, discrete and boolean semantics. In Section 5, we will come back to it however for comparing the analysis of the influence graph between molecules obtained from the differential semantics, with the one obtained from the syntax of the reaction rules, and for establishing equivalence results under some general conditions on the kinetics.

4. A type system for protein functions

In this section, we investigate the use of types for formally relating information on the biological *function* of some proteins to reaction models. For the sake of simplicity, we restrict ourselves to two simple enzymatic functions: kinase and phosphatase. These functions correspond to the action of adding (resp. removing) a phosphate group to (resp. from) a

compound with a covalent binding. We do not consider other categories such as protease in degradation rules, nor acetylase and deacetylase in modification rules, etc. This choice is in accordance with the BIOCHAM syntax which permits to mark the sites of a protein where a group is added, with the operator ~, as in `P~{p,q}` where protein `P` is modified on its sites `p` and `q`, without distinguishing however between phosphorylation, acetylation, methylation, ubiquitination, etc. We thus consider BIOCHAM models containing compounds with different levels of phosphorylation or acetylation, etc. without distinguishing the different forms of modification, and call them phosphorylation by abuse of terminology.

The inference of protein functions in a reaction model is interesting for several reasons. First, the kind of information (kinase activity) collected on proteins can be checked using online databases like for instance GO, the Gene Ontology [1]. Second, in the context of the machine learning techniques implemented in BIOCHAM for completing or revising a model w.r.t. a temporal logic specification [3], the information that an enzyme acts as a kinase or as a phosphatase drastically reduces the search space for adding reactions, and helps to directly find rules and model revisions that are biologically plausible.

4.1. Abstract domain of protein functions

Definition 14. Let kinase(A, B) and phosphatase(A, B) be relations in $\mathcal{M} \times \mathcal{M}$ denoting the kinase (resp. phosphatase) function of A on B. The abstract domain of *protein functions* $\mathcal{D}_{\mathcal{F}} = \mathcal{P}(\{\text{kinase(A, B)} \mid A, B \in \mathcal{M}\} \cup \{\text{phosphatase(A,B)} \mid A, B \in \mathcal{M}\})$ is the powerset of these expressions, ordered by inclusion.

The abstraction function from the syntactical domain, $\alpha_{\mathcal{F}} : \mathcal{D}_{\mathcal{R}} \to \mathcal{D}_{\mathcal{F}}$, associates to a reaction model R the union of the abstractions defined for each single rule and each pair of rules as follows:

$\alpha_{\mathcal{F}}$(`A =[B]=> C`) = {kinase(B,A)} if `C` is more phosphorylated than `A` (i.e. its set of active phosphorylation sites strictly includes that of `A`), in which case *B has kinase function w.r.t. A*;

$\alpha_{\mathcal{F}}$(`A + B => D, D => C + B`) = {kinase(B,A)} if similarly `C` is more phosphorylated than `A`;

$\alpha_{\mathcal{F}}$(`A =[B]=> C`) = {phosphatase(B,A)} if, on the contrary, `A` is more phosphorylated than `C`;

$\alpha_{\mathcal{F}}$(`A + B => D, D => C + B`) = {phosphatase(B,A)} if `A` is more phosphorylated than `C`.

Note that as the abstraction function is not defined pointwise but also on pairs of reaction rules, the time complexity for computing the set of protein functions from the reactions is quadratic in the number of rules. One can easily check that:

Proposition 15. *Let* $\gamma_{\mathcal{F}}(f) = \cup {\alpha_{\mathcal{F}}}^{-1}(\downarrow f)$, $\mathcal{D}_{\mathcal{R}} \underset{\gamma_{\mathcal{F}}}{\overset{\alpha_{\mathcal{F}}}{\longleftarrow\!\!\longrightarrow}} \mathcal{D}_{\mathcal{F}}$ *is a Galois connection.*

This typing for protein functions is very precise as it refers to particular molecules. On the other hand, keeping only the kinase or phosphatase function in an unary predicate without the information on the transformed molecules might be too loose. Between these two extreme choices, one could also consider a hierarchical type structure such as the one defined by the following grammar:

$$\tau ::= \mathit{kinase} | \mathit{phosphatase} | \mathit{kinase}(\tau) | \mathit{phosphatase}(\tau) | T$$

where T denotes some basic types of proteins, with the subtyping relations $\mathit{kinase}(\tau) \preceq \mathit{kinase}$ and $\mathit{phosphatase}(\tau) \preceq \mathit{phosphotase}$. This kind of typing relation stems from models like the MAPK cascade shown in next section where the common denomination for the function of MEK is "MAPK kinase" (i.e. *kinase*(*MAPK*)) and that of RAF is "MAPK kinase kinase" (i.e. *kinase* (*kinase*(*MAPK*))). It is worth noting that such typings are supported by type systems already defined for rule based languages as in [18], using solvers for subtyping constraints in general ordering structures such as quasi-lattices [11] for instance. These considerations are however beyond the scope of this paper and will not be further developed here.

4.2. Evaluation results

4.2.1. MAPK model

On a simple example of the MAPK cascade originally based on [31] and imported into BIOCHAM, the type inference algorithm determines that `RAFK`, `RAF~{p1}` and `MEK~{p1,p2}` have a kinase function; `RAFPH`, `MEKPH` and `MAPKPH` have a phosphatase function; and the other compounds have no function inferred.

If the family of MAPK molecules is given as a basic type, one would moreover infer that the active form of `MEK` is a MAPKK (a kinase for the MAPK family), and that the active form of `RAF` is a MAPKKK (a MAPKK kinase).

If we wanted to type-check such a model, we would correctly check all phosphatases but would miss an example of the kinase function of `MAPK~{p1,p2}`, since its action is not visible in the above model.

4.2.2. Kohn's map

Kohn's map of the mammalian cell cycle control [30] has been transcribed in BIOCHAM to serve as a large benchmarking example of 500 species and 800 rules [8]. This example shows that this abstraction scales up efficiently as the computation of influences requires less than one second CPU time (on a PC 1,7 GHz) in this model. Here is an excerpt of the output of the type inference, where it was restricted to the unary functions *kinase* and *phosphatase* as explained at the end of Section 4.1:

```
cdk7-cycH is a kinase
Wee1 is a kinase
Myt1 is a kinase
cdc25C~{p1} is a phosphatase
cdc25C~{p1,p2} is a phosphatase
Chk1 is a kinase
C-TAK1 is a kinase
Raf1 is a kinase
cdc25A~{p1} is a phosphatase
cycA-cdk1~{p3} is a kinase
cycA-cdk2~{p2} is a kinase
cycE-cdk2~{p2} is a kinase
cdk2~{p2}-cycE~{p1} is a kinase
cycD-cdk46~{p3} is a kinase
cdk46~{p3}-cycD~{p1} is a kinase
cycA-cdk1~{p3} is a kinase
cycB-cdk1~{p3} is a kinase
cycA-cdk2~{p2} is a kinase
cycD-cdk46~{p3} is a kinase
cdk46~{p3}-cycD~{p1} is a kinase
Plk1 is a kinase
pCAF is a kinase
p300 is a kinase
HDAC1 is a phosphatase
```

On the other hand, in these results no compound is both a kinase and a phosphatase. The protein `cdc25A`, `cdc25C` and `HDAC1` are the only phosphatases found in the whole map. The type inference also tells us that the cyclin-dependant kinases have a kinase function when in complex with a cyclin, which is correct. Finally the acetylases `pCAF`, `p300` and the deacetylase `HDAC1` are detected but as expected identified to kinases and phosphatases respectively, since the BIOCHAM syntax does not distinguish between phosphorylation and acetylation.

5. A type system for activation and inhibitory influences

5.1. Abstract domain of influences

Influence networks for activation and inhibition have been introduced for the analysis of gene expression in the setting of gene regulatory networks [41], they basically define graphs where vertices are genes and oriented edges are labelled either with *activates* or *inhibits*, representing the supposed regulation of one gene by another one. Such influence networks are in fact an abstraction of complex reaction networks, and can be applied as such to protein interaction networks. However the distinction between the influence network and the reaction network is crucial for the application of Thomas's conditions of multistationarity and oscillations [41,40] to protein interaction networks, and there has been some confusion between the two kinds of networks [33]. Here we precisely define influence networks as an abstraction (a type system) of reaction networks.

Definition 16. The abstract domain of influences is the powerset of the binary relations of activation and inhibition between compounds $\mathcal{D}_{\mathcal{I}} = \mathcal{P}(\{A \text{ activates } B \mid A, B \in \mathcal{M}\} \cup \{A \text{ inhibits } B \mid A, B \in \mathcal{M}\})$, ordered by inclusion.

5.2. Abstraction from the syntax of the reaction rules

Definition 17. The influence abstraction $\alpha_{\mathcal{RI}} : \mathcal{D}_{\mathcal{R}} \to \mathcal{D}_{\mathcal{I}}$ is the function

$$\begin{aligned}\alpha_{\mathcal{RI}}(x) &= \{A \text{ activates } B \mid \exists (e_i \text{ for } l_i \Rightarrow r_i) \in x,\ l_i(A) > 0 \text{ and } r_i(B) - l_i(B) > 0\} \\ &\cup \{A \text{ inhibits } B \mid \exists (e_i \text{ for } l_i \Rightarrow r_i) \in x,\ l_i(A) > 0 \text{ and } r_i(B) - l_i(B) < 0\}.\end{aligned}$$

In particular, we have the following influences for elementary reactions of complexation, modification, synthesis and degradation:

$\alpha_{\mathcal{RI}}(\{\mathtt{A + B => C}\}) = \{A \text{ inhibits } B, A \text{ inhibits } A, B \text{ inhibits } A, B \text{ inhibits } B, A \text{ activates } C, B \text{ activates } C\}$

$\alpha_{\mathcal{RI}}(\{\mathtt{A = [C] => B}\}) = \{C \text{ inhibits } A, A \text{ inhibits } A, A \text{ activates } B, C \text{ activates } B\}$

$\alpha_{\mathcal{RI}}(\{\mathtt{A = [B] => \_}\}) = \{B \text{ inhibits } A, A \text{ inhibits } A\}$

$\alpha_{\mathcal{RI}}(\{\mathtt{\_ = [B] => A}\}) = \{B \text{ activates } A\}.$

The inhibition loops on the reactants are justified by the negative sign in the differential semantics of the reactions (see Theorem 21 of the next section). These loops are however often omitted in the influence graphs considered in the literature, together with some other influences, according to functionality, kinetics and non-linearity considerations [29].

The abstraction function $\alpha_{\mathcal{R}\mathcal{I}}$ allows us either to type check a reaction model w.r.t. a given influence typing of molecules, or to infer the influence types from the reaction rules. As $\alpha_{\mathcal{R}\mathcal{I}}$ is defined pointwise, it can be computed very efficiently in linear time, and we have by Lemma 2:

Proposition 18. *Let* $\gamma_{\mathcal{R}\mathcal{I}}(f) = \cup{\alpha_{\mathcal{R}\mathcal{I}}}^{-1}(\downarrow f)$, $\mathcal{D}_{\mathcal{R}} \underset{\gamma_{\mathcal{R}\mathcal{I}}}{\overset{\alpha_{\mathcal{R}\mathcal{I}}}{\longleftrightarrow}} \mathcal{D}_{\mathcal{I}}$ *is a Galois connection.*

5.3. Abstraction from the differential semantics of reaction rules

In the differential semantics of a reaction rule model $\{e_i \texttt{ for } l_i\texttt{=>}r_i \mid i \in I\}$ we have $\dot{x_k} = \mathrm{d}x_k/\mathrm{d}t = \sum_{i=1}^{n}(r_i(x_k) - l_i(x_k)) * e_i$. The Jacobian matrix J is formed of the partial derivatives $J_{ij} = \partial\dot{x_i}/\partial x_j$, and one can define the domain $\mathcal{D}_{\mathcal{J}}$ of Jacobians ordered by the pointwise inclusion of codomains. Let us denote by β the mapping from $\mathcal{D}_{\mathcal{R}}$ to $\mathcal{D}_{\mathcal{J}}$ that extracts $\dot{x_k}$ (by the equation given at the beginning of this paragraph) and hence the Jacobian from the kinetic expressions in the reaction rules.

Definition 19. The differential influence abstraction $\alpha_{\mathcal{J}\mathcal{I}} : \mathcal{D}_{\mathcal{J}} \to \mathcal{D}_{\mathcal{I}}$ is the function

$$\begin{aligned}\alpha_{\mathcal{J}\mathcal{I}}(x) &= \{A \text{ activates } B \mid \partial\dot{x_B}/\partial x_A > 0 \text{ in some point of the space}\}\\ &\cup \{A \text{ inhibits } B \mid \partial\dot{x_B}/\partial x_A < 0 \text{ in some point of the space}\}.\end{aligned}$$

The comparison between the differential influences, represented by the function $\alpha_{\mathcal{J}\mathcal{I}} \circ \beta$, and the syntactical influences, represented by the abstraction function $\alpha_{\mathcal{R}\mathcal{I}}$, requires that the information in the kinetic expressions and in the reaction rules are compatible. This motivates the following definition where, intuitively, the first property forbids the absence of purely kinetic inhibitors not represented in the rules, and the second property enforces that reactants and enzymes do appear in rules where they are used.

Definition 20. In a reaction model $x = \{e_i \texttt{ for } l_i\texttt{=>}r_i \mid i \in I\}$, we say that a kinetic expression e_i is *monotonic* iff for all molecules x_k we have

(1) $\partial e_i/\partial x_k \geq 0$ in all points of the space,
(2) $l_i(x_k) > 0$ whenever $\partial e_i/\partial x_k > 0$ in some point of the space.

A reaction model x has a *monotonic kinetics* iff all its reaction rules have monotonic kinetics.

Note that the mass action law kinetics, $e_i = k * \Pi {x_i}^{l_i}$, are monotonic and that Hill's kinetics (of which Michaelis–Menten kinetics are a special case with $n = 1$) $e_i = V_m * {x_s}^n/(K_m + {x_s}^n)$ where $V_m = k * (x_e + x_e * x_s/K_m)$ for an enzymatic reaction $x_s = [x_e] \texttt{=>} x_p$, are also monotonic.[1] On the other hand, inhibitions with negative Hill kinetics of the form $e_i = V_m/(K_m + {x_s}^n)$ are not monotonic, and are not reflected in the syntax of the reactants of the rules.

Theorem 21. *For any reaction model x with monotonic kinetics,* $\alpha_{\mathcal{J}\mathcal{I}} \circ \beta(x) \subseteq \alpha_{\mathcal{R}\mathcal{I}}(x)$.

Proof. If $(A \text{ activates } B) \in \alpha_{\mathcal{J}\mathcal{I}} \circ \beta(x)$ then $\partial\dot{B}/\partial A > 0$. Hence there exists a term in the differential semantics, of the form $(r_i(B) - l_i(B)) * e_i$ with $\partial e_i/\partial A$ of the same sign as $r_i(B) - l_i(B)$.

Let us suppose that $r_i(B) - l_i(B) > 0$ then $\partial e_i/\partial A > 0$ and since e_i is monotonic we get that $l_i(A) > 0$ and thus that $(A \text{ activates } B) \in \alpha_{\mathcal{R}\mathcal{I}}(x)$. If on the contrary $r_i(B) - l_i(B) < 0$ then $\partial e_i/\partial A < 0$, which is not possible for a monotonic kinetics.

If $(A \text{ inhibits } B) \in \alpha_{\mathcal{J}\mathcal{I}} \circ \beta(x)$ then $\partial\dot{B}/\partial A < 0$. Hence there exists a term in the differential semantics, of the form $(r_i(B) - l_i(B)) * e_i$ with $\partial e_i/\partial A$ of sign opposite to that of $r_i(B) - l_i(B)$.

Let us suppose that $r_i(B) - l_i(B) > 0$ then $\partial e_i/\partial A < 0$, which is not possible for a monotonic kinetics. If on the contrary $r_i(B) - l_i(B) < 0$ then $\partial e_i/\partial A > 0$ and since e_i is monotonic we get that $l_i(A) > 0$ and thus that $(A \text{ activates } B) \in \alpha_{\mathcal{R}\mathcal{I}}(x)$. □

It is worth noticing that even in the simple case of mass action law kinetics, there is no equality between $\alpha_{\mathcal{J}\mathcal{I}} \circ \beta$ and $\alpha_{\mathcal{R}\mathcal{I}}$. For instance let x be the following model:

$$\begin{aligned}&k_1 * A \quad \text{for } A \texttt{=>} B\\ &k_2 * A \quad \text{for } \_ = [A] \texttt{=>} A.\end{aligned}$$

We have $\alpha_{\mathcal{R}\mathcal{I}}(x) = \{\text{A activates B, A activates A, A inhibits A}\}$, however $\dot{A} = (k_2 - k_1) * A$, hence $\partial\dot{A}/\partial A$ can be made always positive or always negative or always null, resulting in the absence from $\alpha_{\mathcal{J}\mathcal{I}} \circ \beta(x)$ of, respectively, A inhibits A, A activates A or both.

[1] $x_e * x_s/K_m$ is the concentration of the enzyme-substrate complex, supposed constant in the Michaelian approximation and $x_e + x_e * x_s/K_m$ is thus the total amount of enzyme.

Actually in the general case, β is not monotonic since adding rules can compensate an existing rule in the differential expression and eliminate terms in the differential equations. The differential semantics is thus not an abstraction of the reaction models ordered by set inclusion in the formal sense of abstract interpretation. The above case shows that $\alpha_{\mathcal{J}\mathcal{I}} \circ \beta$ applied to the first rule contains *A inhibits A*, whereas its application to the set of two rules (greater in $\mathcal{D}_{\mathcal{R}}$) may not. A sufficient condition for β to be monotonic is that in the model no kinetic expression can compensate another one in the Jacobian. That is: $\forall x_i, x_j \exists ?k$ s.t. $r_k(x_i) \neq l_k(x_i)$ and $\partial e_k/\partial x_j \neq 0$. This condition is used in the forthcoming Corollary 25. Furthermore, under some hypotheses about the adequateness between the kinetic expressions and the rules, shown to be quite general in the following section, the equality holds between both abstractions.

Definition 22. In a reaction model $x = \{e_i \texttt{ for } l_i \texttt{=>} r_i \mid i \in I\}$, a kinetic expression e_i is *strongly monotonic* iff for all molecules x_k we have

(1) $\partial e_i/\partial x_k \geq 0$ in all points of the space,
(2) $l_i(x_k) > 0$ iff there exists a point in the space s.t. $\partial e_i/\partial x_k > 0$

A reaction model x has a *strongly monotonic kinetics* iff all its reaction rules have a strongly monotonic kinetics.

Note that *strongly monotonic* implies *monotonic*.

Proposition 23. *Mass action law, Michaelis Menten, and Hill kinetics are strongly monotonic.*

Proof. For the case of Mass action law, the kinetics are of the form:

$$e_i = k_i \prod_{l=1}^{m} x_l^{l_i(x_l)}$$

with $k_i > 0$ and $l_i(x_l) \geq 0$. We thus have $\partial e_i/\partial x_k = 0$ if $l_i(x_k) = 0$ and $\partial e_i/\partial x_k = k_i * l_i(x_k) * x_k^{l_i(x_k)-1} \prod_{l \neq k} x_l^{l_i(x_l)}$ otherwise, which clearly satisfies (1) and (2).

In the case of Hill kinetics (of which Michaelis Menten is a subcase), we have:

$$e_i = \frac{V_m * x_s^n}{K_m^n + x_s^n}$$

for the reaction $x_s + x_e \Rightarrow x_p + x_e$ and where $V_m = k_2 * x_e^{tot} = k_2 * (x_e + k_1 * x_e * x_s/(k_{-1} + k_2))$ from the steady state approximation. It is obvious that $\partial e_i/\partial x_k = 0$ for all x_k other than x_s and x_e since they do not appear in e_i and one can easily check that with all the constants n, k_1, k_{-1}, k_2 strictly positive, both $\partial e_i/\partial x_e$ and $\partial e_i/\partial x_s$ are greater than 0 at some point in the space. □

Lemma 24. *Let x be a reaction model with strongly monotonic kinetics, and A and B be two molecules.*
If (A activates B) is in $\alpha_{\mathcal{R}\mathcal{I}}(x)$ but (A inhibits B) is not in $\alpha_{\mathcal{R}\mathcal{I}}(x)$ then (A activates B) is in $\alpha_{\mathcal{J}\mathcal{I}} \circ \beta(x)$.
If (A inhibits B) is in $\alpha_{\mathcal{R}\mathcal{I}}(x)$ but (A activates B) is not in $\alpha_{\mathcal{R}\mathcal{I}}(x)$ then (A inhibits B) is in $\alpha_{\mathcal{J}\mathcal{I}} \circ \beta(x)$.

Proof. Since $\partial \dot{B}/\partial A = \sum_{i=1}^{n} (r_i(B) - l_i(B)) * \partial e_i/\partial A$ and all e_i are monotonic we get that $\partial \dot{B}/\partial A = \sum_{\{i \leq n | l_i(A) > 0\}} (r_i(B) - l_i(B)) * \partial e_i/\partial A$.

Now if (A activates B) is in $\alpha_{\mathcal{R}\mathcal{I}}(x)$ but (A inhibits B) is not in $\alpha_{\mathcal{R}\mathcal{I}}(x)$ then all rule such that $l_i(A) > 0$ verify $r_i(B) - l_i(B) \geq 0$ and there is at least one rule for which the inequality is strict. We thus get that $\partial \dot{B}/\partial A$ is a sum of positive numbers, amongst which one is such that $r_i(B) - l_i(B) > 0$ and $l_i(A) > 0$ which, since x is strongly monotonic, implies that there exists a point in the space for which $\partial e_i/\partial A > 0$ thus $\partial \dot{B}/\partial A > 0$ at that point and (A activates B) is in $\alpha_{\mathcal{J}\mathcal{I}} \circ \beta(x)$.

For inhibition the same reasoning applies with the opposite sign for the $r_i(B) - l_i(B)$ and thus for the finale partial derivative. □

This lemma establishes the following equivalence result:

Theorem 25. *Let x be a reaction model with strongly monotonic kinetics and where no molecule is at the same time an activator and an inhibitor of the same target molecule, then $\alpha_{\mathcal{R}\mathcal{I}}(x) = \alpha_{\mathcal{J}\mathcal{I}} \circ \beta(x)$.*

This theorem shows that for standard kinetic expressions, the syntactical influences coincide with the differential influences based on the signs of the coefficients in the Jacobian matrix, when no molecule is at the same time an activator and an inhibitor of the same molecule. The theorem thus provides a linear time algorithm for computing the differential influences in these cases, simply by computing the syntactical influences. It shows also that the graph of differential influences is independent of the kinetic expressions:

Corollary 26. *The graph of differential influences of a reaction model of n rules with strongly monotonic kinetics is computable in time $O(n)$ if no molecule is at the same time an activator and an inhibitor.*

Corollary 27. *The graph of differential influences of a reaction model is independent of the kinetic expressions as long as they are strongly monotonic, if no molecule is at the same time an activator and an inhibitor.*

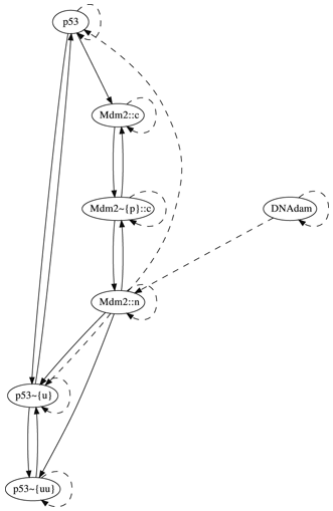


Fig. 1. Inferred influence graph of the p53-Mdm2 model.

5.4. Evaluation results

5.4.1. MAPK model

Let us first consider the MAPK signalling model of [31]. Fig. 6 depicts the reaction graph as a bipartite graph with round boxes for molecules and rectangular boxes for rules. Fig. 7 depicts the inferred influence graph, where activation (resp. inhibition) is materialized by plain (resp. dashed) arrows. The graph layouts of the figures have been computed in BIOCHAM by the Graphviz suite.[2]

Since this model verifies the hypotheses of Corollary 25 we know that abstracting from the kinetics would give the same result.

Interestingly, this influence graph of the MAPK cascade exhibits inhibition feedback loops although in this model, the reaction graph is a pure cascade containing no feedback reaction. The interpretation of these inhibition feedback loops by sequestration in complexes at the different levels of the cascade is analyzed in [43]. The possibility to obtain (damped) oscillations in such "cascades" has been observed in [34] showing the relevance of our automatic analysis in this example.

5.4.2. p53-Mdm2 model

In the p53-Mdm2 model of [10], the protein *Mdm*2 is localized explicitly in two possible locations: the nucleus and in the cytoplasm, and transport rules are considered. Fig. 2 depicts the reaction graph of the model.

Fig. 1 depicts the inferred influence graph. Note that *Mdm*2 in the nucleus has both an activation and an inhibitory effect on $p53 \sim \{u\}$. This corresponds to different influences in different regions of the space and one can check that the two influences also appear in $\alpha_{\mathcal{J}\mathcal{I}}$.

Fig. 3 depicts the core influence graph considered for the logical analysis of this model [29]. In the core influence graph, some influence are neglected, as expected, however some inhibitions, such the inhibitory effect of *p*53 on *Mdm*2 in the nucleus, are considered while they do not appear in the inferred influence graph. The reason for these omissions is the way the reaction model is written. Some inhibitory effects are indeed expressed in the kinetic expression by subtraction of, or division by, the molecular concentration of some compounds that do not appear in the rule itself. Those non-monotonic inhibitions are thus missed by the type inference algorithm. An example of such a rule is the following one for the inhibition of *Mdm*2 by *p*53:

```
macro(p53tot,[p53]+[p53~{u}]+[p53~{uu}]).

(kph*[Mdm2::c]/(Jph+p53tot),MA(kdeph))
   for Mdm2::c <=> Mdm2~{p}::c.
```

Obviously, one cannot expect to infer such inhibitory effects from the reaction rules. Such a situation suggests to extend the syntax of reaction rules in order to indicate the inhibitors of the reaction, in a somewhat symmetric fashion to what is done for catalysts.

[2] http://www.graphviz.org/.

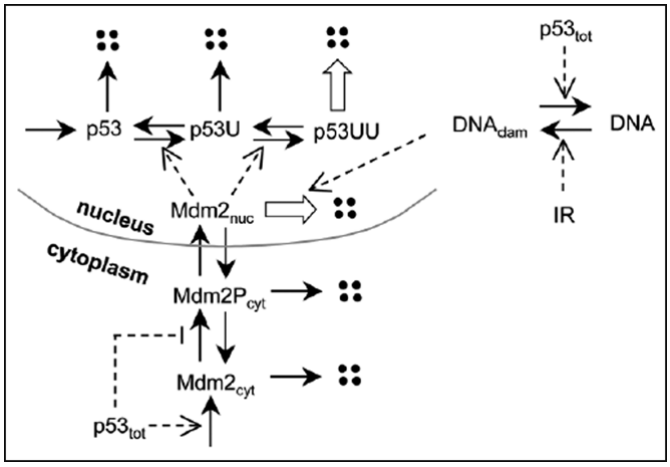


Fig. 2. Original reaction graph considered in [10] for the p53-Mdm2 model.

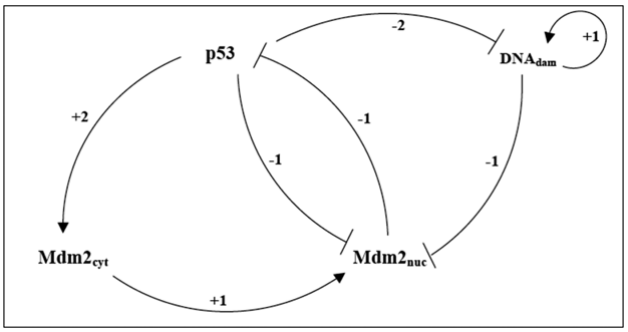


Fig. 3. Core influence graph [29].

5.4.3. Kohn's map

On the quite big model of Kohn's map, the type inference of activation and inhibition influences from reaction rules takes less than one second CPU time (on a PC 1,7 GHz) for the complete model, showing again the efficiency of the type inference algorithm.

As kinetic data is typically missing for such a large model, the influence analysis from the syntactical domain is the only one available.

6. A type system for location topologies

To date, models of biochemical systems generally abstract from space considerations. Models taking into account cell compartments and transport phenomena are thus much less common. Nevertheless, with the advent of systems biology computational tools, more and more models are refined with space considerations and transport delays, e.g. [10]. In SBML [26] level 1 version 1, locations have been introduced as purely symbolic compartments without precise topology. We show in this section how the topology can be inferred from the reaction rules, and checked in different models.

6.1. Abstract domain of location topologies

We will now focus on the notion of *neighbor* that is supposed to represent the fact that two compounds live in two compartments that are next to each other. In SBML level 2, an *outside* relation can optionally be given for two compartments, stating that one is the outside of the other one. We should have, from our definition, that if A is the outside of B, then any compound living in A and any compound living in B are neighbors.

Definition 28. The domain of neighborhood relations $\mathcal{D}_{\mathcal{N}} = \mathcal{P}(\mathcal{M} \times \mathcal{M})$ is about pairs of molecules. $\alpha_{\mathcal{N}} : \mathcal{D}_{\mathcal{R}} \to \mathcal{D}_{\mathcal{N}}$ is defined by the union of its definition on single rules:

$\alpha_{\mathcal{N}}(\texttt{E for } \texttt{A}_1 + \cdots + \texttt{A}_n \texttt{ => } \texttt{B}_1 + \cdots + \texttt{B}_m) =$ All A_i and all B_j are pairwise neighbors, and for all C_k such that $\texttt{[C}_k\texttt{]}$ appears in $\texttt{E}$, C_k is a neighbor of all A_i and all B_j.

Proposition 29. *Let* $\gamma_{\mathcal{N}}(n) = \cup {\alpha_{\mathcal{N}}}^{-1}(\downarrow n)$. $\mathcal{D}_{\mathcal{R}} \underset{\gamma_{\mathcal{N}}}{\overset{\alpha_{\mathcal{N}}}{\longleftrightarrow}}$ *is a Galois connection.*

The concretization of a positive neighborhood between two locations is the set of all possible rules linking those compartments, i.e. transport rules or rules influencing one compartment from another one. It describes in some sense the interface between the two locations.

6.2. Evaluation results

6.2.1. Models from `biomodels.net`

We have taken models from the literature through the http://www.biomodels.net database. Of the 112 curated models in the current version (dated June 2007) only 35 have more than one compartment, and only 7 of those use the *outside* attribute of SBML to provide more topological insight.

The neighborhood relation is inferred in these models imported in BIOCHAM, and then checked consistent with the provided *outside* relation.

For instance for calcium oscillations, we tried both the Marhl et al. model of [32] and the Borghans et al. model of [2].

In the first case (model `BIOMD0000000039.xml`), three locations are defined: the cytosol, the endoplasmic reticulum and a mitochondria, from the reactions the inferred topology is that the cytosol is neighbor of the two other locations. This correspond exactly to the information obtained from the *outside* annotations (the cytosol being marked as the outside of the two other locations).

In the second case (models `BIOMD0000000043.xml` to `BIOMD0000000045.xml`) we focused on the last model (*two-pool*) since it is the only one with 4 different locations: the extracellular space, the cytosol and two internal vesiculae. The location inference produces a topology where the cytosol is neighbor of all other locations. Once again this is correct w.r.t. the outside information provided in the SBML file: both vesiculae have the cytosol as outside location and the cytosol itself has the extracellular space as outside location.

These considerations show that there is some mismatch between the SBML reaction models and the choice of expressing outside vs neighborhood properties of locations. In the perspective of type checking and type inference, neighborhood relations should be preferred as they can be checked, or inferred from the reaction model, whereas the outside relation contain more information that, while helpful for the modeler as meta-data, cannot be handled automatically without abstracting it first in neighbors properties. Note however that the SBML v. 3 effort rather goes in the opposite direction w.r.t. spatial information (see http://sbml.org/wiki/Spatial_Features) since it will allow a complete geometrical description of the compartments, which is of course very informative but is not amenable to automatic checking or inference.

Note also that in calculi where the topology of the network evolves, like the Brane calculus [6] and its derivatives, the outside and inside relationships change much more radically than the neighborhood relationship. For instance an exocytosis followed by an endocytosis might reverse the outside relationship whereas it would not change the neighborhood relation. Moreover, as shown in the second example below the neighborhood relation can easily be applied to cell (or compartment) populations to represent the topology, while defining only one "outside" for each cell makes the topology disappear.

6.2.2. P53/Mdm2

The first example comes from [10]: a model of the p53/Mdm2 interaction with two locations (see Fig. 2) where the transport between cytoplasm and nucleus is necessary to explain some time delays observed in the mutual repression of these proteins.

```
biocham: load_biocham('EXAMPLES/locations/p53Mdm2.bc').
...
(MA(ko),MA(ki)) for Mdm2::n <=> Mdm2~{p}::c.
...
biocham: show_neighborhood.
c and n are neighbors
```

We restricted the output to the neighborhood between compartments rather than compounds for clarity.

In this precise case, the model as published does not systematically use the volume ratio in the kinetics. The transcription and type-checking of the model showed that if one wanted to keep the background degradation rate of *Mdm*2 (without DNA damage) independent of the location, one obtains different kinetics than those of the published model. In this case a formal transcription in BIOCHAM (or SBML) provided a supplementary model-validation step.

6.2.3. Delta and notch model

The Delta and Notch proteins are crucial to the cell fate in different organisms. A population of neighboring cells is represented through locations, chosen here to be on a square grid. The model of Gosh and Tomlin [20] for the activation and inhibition of Delta and Notch proteins reproduce the salt-and-pepper coloring of the cells corresponding to high Delta-low Notch and low Delta-high Notch differentiation. This is typical of the Delta-Notch lateral inhibition based differentiation. The signaling pathways are simplified to the extreme to take into account only the direct effect of Delta and Notch expression in the cell and on the neighboring cells, with rules like:

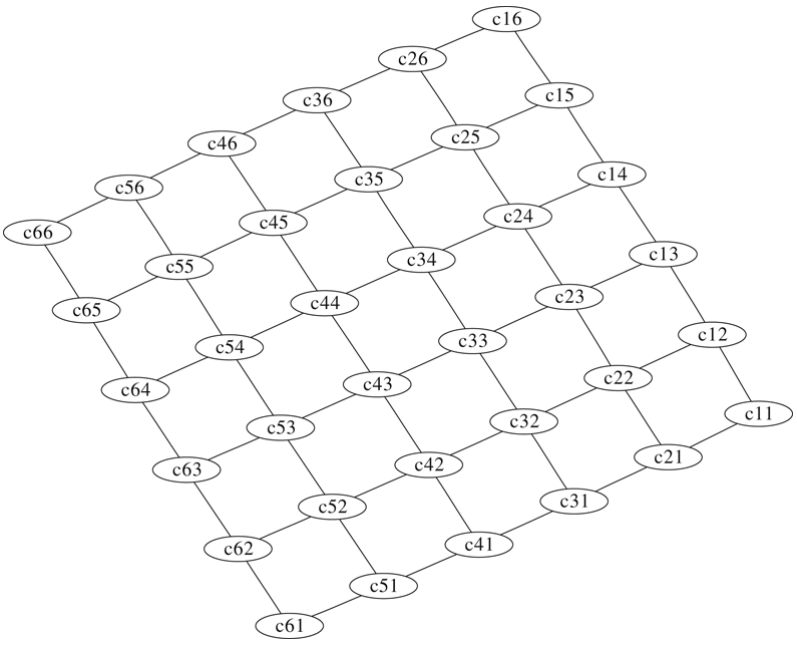


Fig. 4. Delta-Notch square cell grid inferred by $\alpha_{\mathcal{N}}$ in a 6x6 model.

```
biocham: load_biocham('EXAMPLES/locations/notch4n36c.bc').

(if [D::c21]+[D::c23]+[D::c12]+[D::c32] < 0.2
   then 0
   else ka,MA(kd)) for
      _ <=> N::c22.

(if [N::c22] > 0.5
   then 0
   else ka,MA(kd)) for
      _ <=> D::c22.

...
```

Note that in this example, as most of the information is in the kinetics of the rules, the analysis of influences should be done with the Jacobian of the differential semantics, instead of the syntactic domain of reaction rules, as described in Section 5. However, for the analysis of location topology, the abstraction defined in this section provides the expected result, as depicted in Fig. 4.

This example also illustrates a subtlety in the definition of the abstraction function $\alpha_{\mathcal{N}}$. Indeed, it could be tempting to define the abstraction in the following simpler manner:

Definition 30. $\alpha'_{\mathcal{N}} : \mathcal{D}_{\mathcal{R}} \to \mathcal{D}_{\mathcal{N}}$ is defined by the union of its definition on single rules:
$\alpha'_{\mathcal{N}}(\texttt{E for A}_1 + \cdots + \texttt{A}_n \texttt{ => B}_1 + \cdots + \texttt{B}_m) =$ All A_i, all B_j, and all C_k such that $\texttt{[C}_k\texttt{]}$ appears in E, are pairwise neighbors.

Fig. 5 depicts the topology inferred for Delta-Notch model with this second definition. It shows too coarse on such examples since co-modifiers are put in the kinetic expression of a single rule for simplification purposes. This illustrates the fact that lots of published reaction models rely extensively on the ODEs derived from the rules, the rules themselves being not always carefully written, but rather as compact as possible.

7. Conclusion

We have shown that the framework of abstract interpretation applies, on the one hand, to the organization of major semantics of biochemical reaction rules into a hierarchy of semantics related by abstraction functions, and on the other hand, to the formalization of some further abstractions commonly used in systems biology as type systems.

In the three type systems studied in this paper for, respectively, protein functions, activation and inhibitory influences, and location topologies, the analyses are based on static information gained directly from the syntax of reaction rules, without considering their formal semantics, nor their precise dynamics. It is worth noting that this situation also occurs in program analysis where the syntax of programs may capture a sufficient part of the semantics for many analyses. Here, it

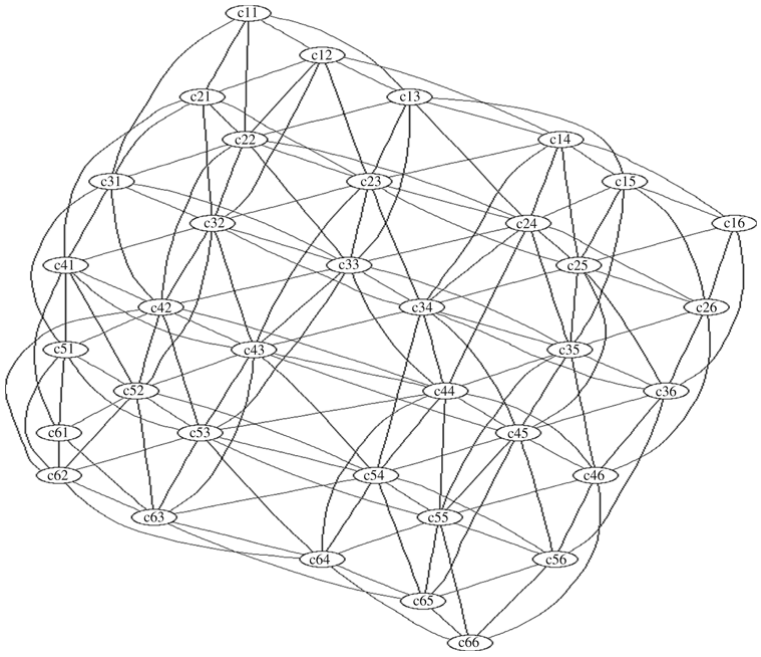


Fig. 5. Delta-Notch square cell grid inferred by $\alpha'_{\mathcal{N}}$ in a 6x6 model.

is remarkable that such simple analyses already provide useful information on biological models, independently from their dynamics for which different definitions are considered (discrete, continuous, stochastic, etc.).

The formal definition of the influence graph as an abstraction of the reaction model eliminates some confusion that exists in the use of Thomas's conditions [41,40] for the analysis of reaction models [33]. Such a formalization shows also that the influence graphs usually considered in the literature are further abstractions obtained by forgetting some influences, based on non-linearity considerations [42]. Some inhibitions may also be missing in the inferred influences when they are hidden in the kinetic expressions of the reactions and do not appear explicitly in the reactants. This suggests either to refine the abstraction function to take into account the kinetic expression when possible, or to extend the syntax of reactions in order to make explicit such inhibitory effects, in a symmetric fashion to catalysts for activations. In SBML there is actually an unique symmetrical notion of *Modifiers* which is not sufficient to infer the influence graph since it does not make any difference between activators and inhibitors.

Furthermore, we have shown that under general monotonicity conditions satisfied by standard kinetics, such as the mass action law, Michaelis–Menten or Hill kinetics, the influences inferred from the syntax of reactants and products in the rules, include the influences inferred from the signs of the coefficient of the Jacobian matrix, and the equality holds when no molecule is both an activator and an inhibitor of a same molecule. This shows, perhaps surprisingly, that the Jacobian influences can be easily computed in linear time from the rule syntax, and that they are independent of the precise kinetic expressions under general conditions.

Similarly, the inference of protein functions and of location neighborhood have shown that the static analysis of reaction models by type inference provides both accurate and useful information. They also provide some guidelines for the extensions of biochemical reaction languages, like for instance in BIOCHAM, differentiating phosphorylation from other forms of modifications like acetylation, methylation, ubiquitination, etc. and in SBML, considering neighborhood rather than outside properties, and introducing a syntax for compound modifications.

Although the analyses done from the differential semantics of reaction rules have been compared to the analyses done from the syntax of reaction rules, the differential semantics itself is the only one that has not been related by Galois connections to the other semantics for several reasons explained in the corresponding section of this paper. These difficulties obviously provide an interesting subject for future work, from both the systems biology and the abstract interpretation theory standpoints.

Acknowledgement

We gracefully acknowledge Monika Heiner for interesting discussions on the modeling of biochemical networks with Petri nets. This work benefited from partial support of the Network of Excellence REWERSE of the European Union and of the INRIA ARC MOCA.

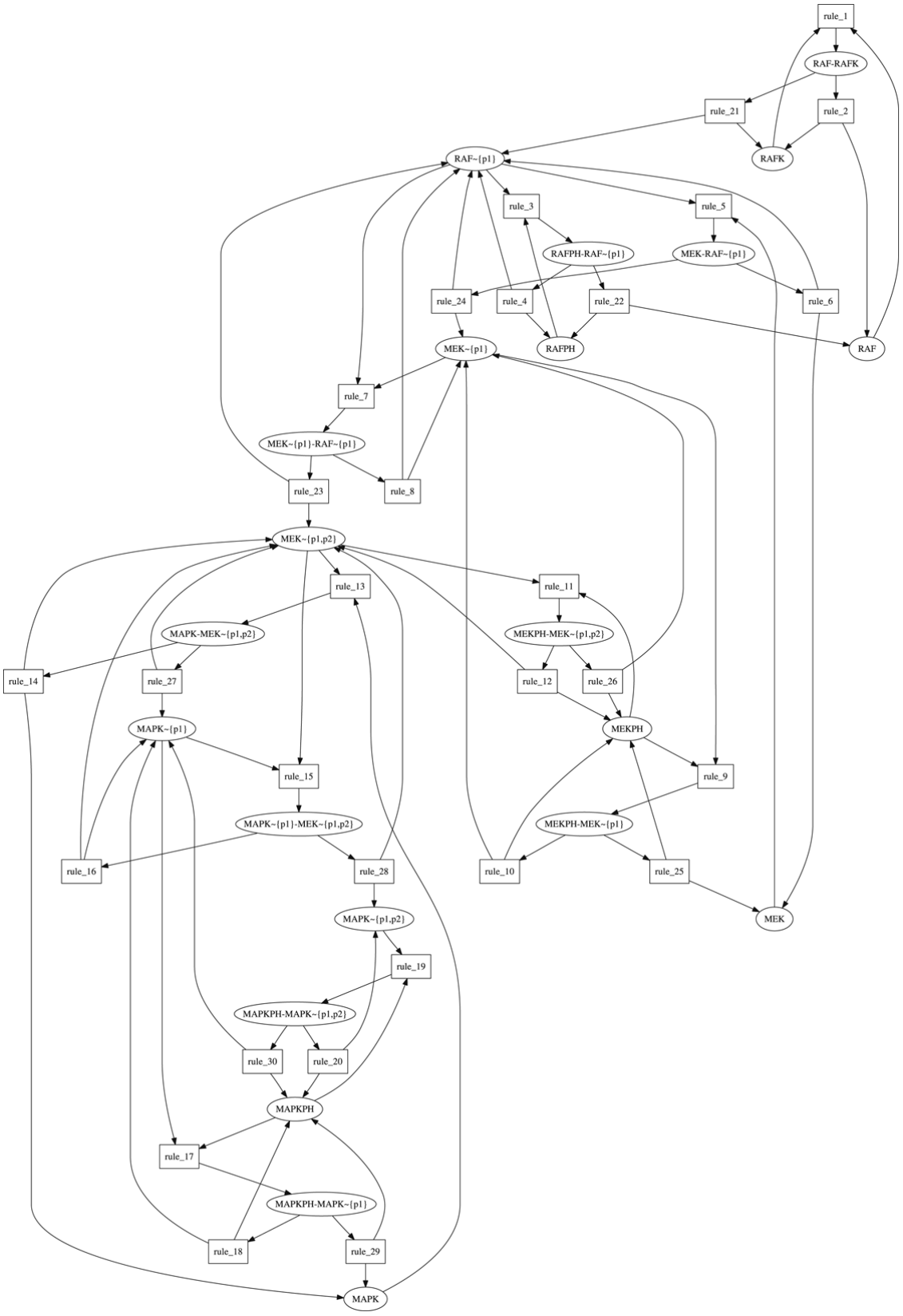


Fig. 6. Reaction graph of the MAPK cascade model.

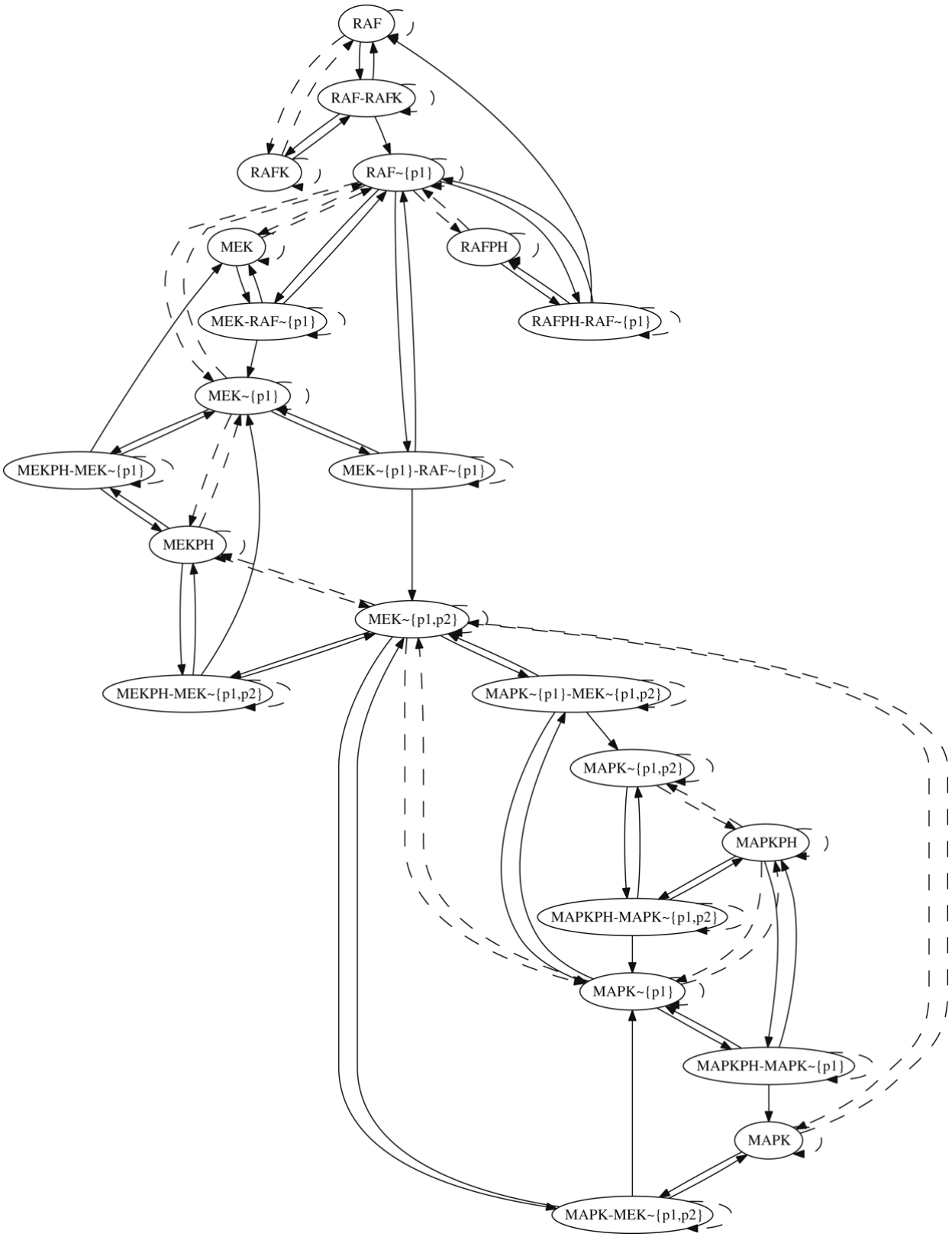


Fig. 7. Inferred influence graph of the MAPK cascade model showing negative feedback.

References

[1] Michael Ashburner, Catherine A. Ball, Judith A. Blake, David Botstein, Heather Butler, J. Michael Cherry, Allan P. Davis, Kara Dolinski, Selina S. Dwight, Janan T. Eppig, Midori A. Harris, David P. Hill, Laurie Issel-Tarver, Andrew Kasarskis, Suzanna Lewis, John C. Matese, Joel E. Richardson, Martin Ringwald, Gerald M. Rubin, Gavin Sherlock, Gene ontology: Tool for the unification of biology, Nature Genetics 25 (2000) 25–29.
[2] José Borghans, Geneviève Dupont, Albert Goldbeter, Complex intracellular calcium oscillations: A theoretical exploration of possible mechanisms, Biophysical Chemistry 66 (1997) 25–41.
[3] Laurence Calzone, Nathalie Chabrier-Rivier, François Fages, Sylvain Soliman, Machine learning biochemical networks from temporal logic properties, in: Gordon Plotkin (Ed.), Transactions on Computational Systems Biology VI, CMSB'05, in: Lecture Notes in BioInformatics, vol. 4220, Springer-Verlag, November 2006, pp. 68–94. Special Issue.
[4] Laurence Calzone, François Fages, Sylvain Soliman, BIOCHAM: An environment for modeling biological systems and formalizing experimental knowledge, BioInformatics 22 (14) (2006) 1805–1807.
[5] Luca Cardelli, Typeful programming, in: E.J. Neuhold, M. Paul (Eds.), Formal Description of Programming Concepts, Springer-Verlag, Berlin, 1991, pp. 431–507.
[6] Luca Cardelli, Brane calculi - interactions of biological membranes, in: Vincent Danos, Vincent Schächter (Eds.), CMSB'04: Proceedings of the Second International Workshop on Computational Methods in Systems Biology, in: Lecture Notes in BioInformatics, vol. 3082, Springer-Verlag, 2004, pp. 257–280.
[7] Nathalie Chabrier, François Fages, Symbolic model cheking of biochemical networks, in: Corrado Priami (Ed.), CMSB'03: Proceedings of the First Workshop on Computational Methods in Systems Biology, in: Lecture Notes in Computer Science, vol. 2602, Springer-Verlag, Rovereto, Italy, March 2003, pp. 149–162.
[8] Nathalie Chabrier-Rivier, Marc Chiaverini, Vincent Danos, François Fages, Vincent Schächter, Modeling and querying biochemical interaction networks, Theoretical Computer Science 325 (1) (2004) 25–44.
[9] Claudine Chaouiya, Petri net modelling of biological networks, Briefings in Bioinformatics (2007).
[10] Andrea Ciliberto, Béla Novák, John J. Tyson, Steady states and oscillations in the p53/mdm2 network, Cell Cycle 4 (3) (2005) 488–493.
[11] Emmanuel Coquery, François Fages, Subtyping constraints in quasi-lattices, in: P. Pandya, J. Radhakrishnan (Eds.), Proceedings of the 23rd Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS'2003, in: Lecture Notes in Computer Science, vol. 2914, Springer-Verlag, Mumbai, India, December 2003, pp. 136–148.
[12] Patrick Cousot, Abstract interpretation and application to logic programs, Journal of Logic Programming 13 (2–3) (1992) 103–179.
[13] Patrick Cousot, Types as abstract interpretation (invited paper), in: POPL'97: Proceedings of the 24th ACM Symposium on Principles of Programming Languages, ACM Press, New York, Paris, 1997, pp. 316–331.
[14] Patrick Cousot, Constructive design of a hierarchy of semantics of a transition system by abstract interpretation, Theoretical Computer Science 277 (1) (2002) 47–103.
[15] Patrick Cousot, Radhia Cousot, Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints, in: POPL'77: Proceedings of the 6th ACM Symposium on Principles of Programming Languages, ACM Press, New York, 1977, pp. 238–252. Los Angeles.
[16] Steven Eker, Merrill Knapp, Keith Laderoute, Patrick Lincoln, José Meseguer, M. Kemal Sönmez, Pathway logic: Symbolic analysis of biological signaling, in: Proceedings of the seventh Pacific Symposium on Biocomputing, January 2002, pp. 400–412.
[17] François Fages, From syntax to semantics in systems biology - towards automated reasoning tools, Transactions on Computational Systems Biology IV 3939 (2006) 68–70.
[18] François Fages, Emmanuel Coquery, Typing constraint logic programs, Journal of Theory and Practice of Logic Programming 1 (6) (2001) 751–777.
[19] François Fages, Sylvain Soliman, Nathalie Chabrier-Rivier, Modelling and querying interaction networks in the biochemical abstract machine BIOCHAM, Journal of Biological Physics and Chemistry 4 (2) (2004) 64–73.
[20] Ronojoy Ghosh, Claire Tomlin, Lateral inhibition through delta-notch signaling: A piecewise affine hybrid model, in: Springer-Verlag, (Ed.), Proceedings of the 4th International Workshop on Hybrid Systems: Computation and Control, HSCC'01, in: Lecture Notes in Computer Science, vol. 2034, Rome, Italy, 2001, pp. 232–246.
[21] Michael A. Gibson, Jehoshua Bruck, A probabilistic model of a prokaryotic gene and its regulation, in: H. Bolouri, J.M. Bower (Eds.), Computational Methods in Molecular Biology: From Genotype to Phenotype, MIT press, 2000 (Chapter 2).
[22] David Gilbert, Monika Heiner, Sebastian Lehrack, A unifying framework for modelling and analysing biochemical pathways using petri nets, in: CMSB'07: Proceedings of the Fifth International Conference on Computational Methods in Systems Biology, in: Lecture Notes in Computer Science, vol. 4695, Springer-Verlag, 2007.
[23] Daniel T. Gillespie, General method for numerically simulating stochastic time evolution of coupled chemical-reactions, Journal of Computational Physics 22 (1976) 403–434.
[24] Daniel T. Gillespie, Exact stochastic simulation of coupled chemical reactions, Journal of Physical Chemistry 81 (25) (1977) 2340–2361.
[25] William S. Hlavacek, James R. Faeder, Michael L. Blinov, Richard G. Posner, Michael Hucka, Walter Fontana, Rules for modeling signal-transduction systems, Science STKE 344 (2006) re6.
[26] Michael Hucka, et al., The systems biology markup language (SBML) a medium for representation and exchange of biochemical network models, Bioinformatics 19 (4) (2003) 524–531.
[27] Trey Ideker, Timothy Galitski, Leroy Hood, A new approach to decoding life: Systems biology, Annual Review of Genomics and Human Genetics 2 (2001) 343–372.
[28] Minoru Kanehisa, Susumu Goto, KEGG: Kyoto encyclopedia of genes and genomes, Nucleic Acids Research 28 (1) (2000) 27–30.
[29] Marcelle Kaufman, Private communication, 2006.
[30] Kurt W. Kohn, Molecular interaction map of the mammalian cell cycle control and DNA repair systems, Molecular Biology of the Cell 10 (8) (1999) 2703–2734.
[31] Andre Levchenko, Jehoshua Bruck, Paul W. Sternberg, Scaffold proteins may biphasically affect the levels of mitogen-activated protein kinase signaling and reduce its threshold properties, PNAS 97 (11) (2000) 5818–5823.
[32] Marko Marhl, Thomas Haberichter, Milan Brumen, Reinhart Heinrich, Complex calcium oscillations and the role of mitochondria and cytosolic proteins, BioSystems 57 (2000) 75–86.
[33] Nick I. Markevich, Jan B. Hoek, Boris N. Kholodenko, Signaling switches and bistability arising from multisite phosphorylation in protein kinase cascades, Journal of Cell Biology 164 (3) (2005) 353–359.
[34] Liang Qiao, Robert B. Nachbar, Ioannis G. Kevrekidis, Stanislav Y. Shvartsman, Bistability and oscillations in the huang-ferrell model of mapk signaling, PLoS Computational Biology 3 (9) (2007) 1819–1826.
[35] V.N. Reddy, M.L. Mavrovouniotis, M.N. Liebman, Petri net representations in metabolic pathways, in: L. Hunter, D.B. Searls, J.W. Shavlik (Eds.), Proceedings of the 1st International Conference on Intelligent Systems for Molecular Biology, ISMB, AAAI Press, 1993, pp. 328–336.
[36] Andrea Sackmann, Monika Heiner, Ina Koch, Application of petri net based analysis techniques to signal transduction pathways, BMC Bioinformatics 7 (482) (2006).
[37] S. Schuster, T. Pfeiffer, F. Moldenhauer, et al., Exploring the pathway structure of metabolism: Decomposition into subnetworks and application to mycoplasma pneumoniae, Bioinformatics 18 (3) (2002) 51–61.
[38] Lee A. Segel, Modeling Dynamic Phenomena in Molecular and Cellular Biology, Cambridge University Press, 1984.

[39] Sylvain Soliman, François Fages, CMBSlib: A library for comparing formalisms and models of biological systems, in: Vincent Danos, Vincent Schächter (Eds.), CMSB'04: Proceedings of the Second International Workshop on Computational Methods in Systems Biology, in: Lecture Notes in BioInformatics, vol. 3082, Springer-Verlag, 2004, pp. 231–235.
[40] Christophe Soulé, Graphic requirements for multistationarity, ComplexUs 1 (2003) 123–133.
[41] René Thomas, Anne-Marie Gathoye, Lucie Lambert, A complex control circuit: regulation of immunity in temperate bacteriophages, European Journal of Biochemistry 71 (1) (1976) 211–227.
[42] René Thomas, Marcelle Kaufman, Multistationarity, the basis of cell differentiation and memory, Chaos 11 (1) (2001) 170–195.
[43] Alejandra C. Ventura, Jacques-Alexandre Sepulchre, Soffia D. Merajver, A hidden feedback in signaling cascades is revealed, PLoS, Computational Biology 4 (3) (2008) e1000041.
[44] I. Zevedei-Oancea, S. Schuster, Topological analysis of metabolic networks based on petri net theory, Silico Biology 3 (29) (2003).

4.1.2 From Reaction Models to Influence Graphs and Back: A Theorem

The case of influence graphs/regulatory networks, i.e., the structural part of a logical model *à la* Thomas, as abstractions of reaction models can be further studied. Indeed, classical reaction models will never represent any direct kinetic inhibition. The only cases where negative influences appear are catalyzed degradation or sequestration, but do not correspond to inhibitory kinetics like $\frac{1}{k+x^n}$. In this article we describe more precisely the inference of influence graphs, extending its soundness and completeness to reaction models with explicit inhibitors (*antagonists*). Note that these results are not only the basis of the following articles in this chapter but are also a prerequisite for our analysis of multistationarity described in Section 2.4.

[7] François Fages and Sylvain Soliman. "From reaction models to influence graphs and back: a theorem". In: *Proceedings of Formal Methods in Systems Biology FMSB'08*. Lecture Notes in Computer Science 5054. Springer-Verlag, Feb. 2008. DOI: 10.1007/978-3-540-68413-8_7

From Reaction Models to Influence Graphs and Back: A Theorem*

François Fages and Sylvain Soliman

Projet Contraintes, INRIA Rocquencourt,
BP105, 78153 Le Chesnay Cedex, France
http://contraintes.inria.fr

Abstract. Biologists use diagrams to represent interactions between molecular species, and on the computer, diagrammatic notations are also more and more employed in interactive maps. These diagrams are fundamentally of two types: reaction graphs and activation/inhibition graphs. In this paper, we study the formal relationship between these graphs. We consider systems of biochemical reactions with kinetic expressions, as written in the Systems Biology Markup Language SBML, and interpreted by a system of Ordinary Differential Equations over molecular concentrations. We show that under a general condition of increasing monotonicity of the kinetic expressions, and in absence of both activation and inhibition effects between a pair of molecules, the influence graph inferred from the stoichiometric coefficients of the reactions is equal to the one defined by the signs of the coefficients of the Jacobian matrix. Under these conditions, satisfied by mass action law, Michaelis-Menten and Hill kinetics, the influence graph is thus independent of the precise kinetic expressions, and is computable in linear time in the number of reactions. We apply these results to Kohn's map of the mammalian cell cycle and to the MAPK signalling cascade. Then we propose a syntax for denoting antagonists in reaction rules and generalize our results to this setting.

1 Introduction

Biologists use diagrams to represent interactions between molecular species, and diagrammatic notations like the ones introduced by Kohn in his map of the mammalian cell cycle [2] are also employed on the computer in interactive maps, like for instance MIM[1]. This type of notation encompasses two types of information : interactions (binding, complexation, protein modification, etc.) and regulations (of an interaction or of a transcription).

The Systems Biology Markup Language (SBML) [3] uses a syntax of reaction rules with kinetic expressions to define reaction models in a precise way, and more and more models are described in such a formalism, like in the biomodels.net

* This paper provides a direct presentation and a generalization of one theorem shown in [1] among other results in the framework of abstract interpretation which is not used here.

[1] http://discover.nci.nih.gov/mim/

J. Fisher (Ed.): FMSB 2008, LNBI 5054, pp. 90–102, 2008.

repository. This type of language is well suited to describe interactions (and in a limited manner their regulations through the notion of *modifiers*) but not directly molecule to molecule activations and inhibitions.

On the other hand, formal influence graphs for activation and inhibition have been introduced in the setting of gene regulatory networks [4] as an abstraction of complex reaction networks. These graphs completely abstract from the precise interactions, especially at post-transcriptional level, and retain only the activation and inhibition effects between genes. In these influence graphs, the existence of a positive circuit (resp. a negative circuit) has been shown to be a necessary condition for multistationarity (resp. oscillations) in different settings [5,6,7,8,9], as conjectured by Thomas [10].

There are nowadays several tools providing different kinds of analyses for either reaction models or influence graphs. However the only formal relationship relating the two seems to be the extraction of the influence graph from the Jacobian matrix derived from the reaction model, when equipped with precise kinetic expressions and parameter values.

In this paper, we study more systematically the formal relationship between reaction models and activation/inhibition influence graphs. We consider systems of biochemical reactions with kinetic expressions, as written in the Systems Biology Markup Language SBML, and interpreted by systems of Ordinary Differential Equations over molecular concentrations. We show that under the general condition of strongly increasing monotonicity of the kinetic expressions, and in absence of both activation and inhibition effects from one molecule to the same target, the influence graph inferred from the stoichiometric coefficients of the reactions, called the syntactical influence graph, is identical to the influence graph defined by the signs of the coefficients of the Jacobian matrix, called the differential influence graph. Under these conditions, satisfied by mass action law, Michaelis-Menten and Hill kinetics, the influence graph is thus independent of the kinetic expressions for the reactions, and is computable in linear time in the number of reactions.

We show that this remarkable property applies to the transcription of Kohn's map of the mammalian cell cycle control [2] into an SBML model of approx. 800 reactions [11]. On this example, the syntactical influence graph is computed in less than one second, and our equivalence theorem shows that this influence graph would be the same as the differential influence graph for any standard kinetics and any (non zero) parameter values. The same property of independence from the kinetic expressions holds for the influence graph inferred from the MAPK signalling model of Levchenko et al. [12]. This influence graph exhibits positive as well as negative feedbacks that are hidden in the purely directional cascade of the reaction graph [13], and that have been the reason for an erroneous interpretation of Thomas' rules when applied to the MAPK cascade in [14].

Finally, we consider generalized reaction rules, where inhibitors can be indicated in the syntax of the rules, and generalize our results to this setting for a large set of kinetic expressions.

2 Reaction Models

Following SBML and BIOCHAM [15,16] conventions, a model of a biochemical system is formally a set of reaction rules of the form e `for` S `=>` S' where S is a set of molecules given with their stoichiometric coefficient, called a *solution*, S' is the transformed solution, and e is a kinetic expression involving the concentrations of molecules (which are not strictly required to appear in S).

We will use the BIOCHAM operators `+` and `*` to denote solutions as `2*A + B`, as well as the syntax of catalyzed reactions `e for S =[C]=> S'` as an abbreviation for `e for S+C => S'+C`.

Classical kinetic expressions are the mass action law kinetics

$$k * \prod_{i=1}^{n} {x_i}^{l_i}$$

for a reaction with n reactants x_i, where l_i is the stoichiometric coefficient of x_i as a reactant, Michaelis-Menten kinetics

$$V_m * x_s/(K_m + x_s)$$

for an enzymatic reaction of the form $x_s = [x_e] => x_p$, where[2] $V_m = k * (x_e + x_e * x_s/K_m)$, and Hill's kinetics

$$V_m * {x_s}^n/(K_m^n + {x_s}^n)$$

of which Michaelis-Menten kinetics is a special case with $n = 1$.

A set of reaction rules $\{e_i$ `for` S_i `=>` $S'_i\}_{i=1,...,n}$ over molecular concentration variables $\{x_1, ..., x_m\}$, can be interpreted under different semantics. The traditional *differential semantics* interpret the rules by the following system of Ordinary Differential Equations (ODE):

$$dx_k/dt = \sum_{i=1}^{n} r_i(x_k) * e_i - \sum_{j=1}^{n} l_j(x_k) * e_j$$

where $r_i(x_k)$ (resp. l_i) is the stoichiometric coefficient of x_k in the right (resp. left) member of rule i.

The differential semantics will be the only interpretation of reaction models considered here. In this paper, we shall not consider the other interpretations of reaction rules used in BIOCHAM [1], namely the *stochastic semantics*, where the kinetic expressions are interpreted as transition probabilities, the rule set as a continuous-time Markov chain that can be simulated with Gillespie's algorithm [17], or the *boolean semantics* which simply forgets the kinetic expressions and interpret the rules as a non-deterministic (asynchronous) transition system over boolean states representing the absence or presence of molecules.

[2] $x_e * x_s/K_m$ is the concentration of the enzyme-substrate complex, supposed constant in the Michaelian approximation and $x_e + x_e * x_s/K_m$ is thus the total amount of enzyme.

3 Influence Graphs of Activation and Inhibition

Influence graphs for activation and inhibition have been introduced for the analysis of gene expression in the setting of gene regulatory networks [4]. Such influence graphs are in fact an abstraction of complex reaction networks, and can be applied as such to protein interaction networks. However the distinction between the influence graph and the reaction (hyper)graph is crucial to the application of Thomas's conditions of multistationarity and oscillations [4,7] to protein interaction network, and there has been some confusion between the two kinds of graphs [14].

Here we consider two definitions of the influence graph associated to a reaction model, and show their equivalence under general assumptions.

3.1 Definition from the Jacobian Matrix

In the differential semantics of a reaction rule model $M = \{e_i \texttt{ for } l_i \texttt{ => } r_i \mid i \in I\}$ we have $\dot{x_k} = dx_k/dt = \sum_{i=1}^{n}(r_i(x_k) - l_i(x_k)) * e_i$. The Jacobian matrix J is formed of the partial derivatives $J_{ij} = \partial \dot{x_i}/\partial x_j$.

Definition 1. *The* differential influence graph *associated to a reaction model is the graph having for vertices the molecular species, and for edge-set the following two kinds of edges:*

$\{A$ *activates* $B \mid \partial \dot{x_B}/\partial x_A > 0$ *in some point of the space*$\}$
$\cup\{A$ *inhibits* $B \mid \partial \dot{x_B}/\partial x_A < 0$ *in some point of the space*$\}$

Both activation and inhibition edges may exist between two molecular species in reaction models such as for instance:

$k_1 * A$ for A `=>` B
$k_2 * A * B$ for $A + B$ `=>` C

We have indeed $dB/dt = k_1 * A - k_2 * A * B$ and $\partial \dot{B}/\partial A = k_1 - k_2 * B$, hence A *inhibits* B and A *activates* B both belong to the differential influence graph in such an example.

3.2 Definition from the Stoichiometric Coefficients

Definition 2. *The* syntactical influence graph *associated to a reaction model* M *is the graph having for vertices the molecular species, and for edges the following set:*

$\{A$ *inhibits* $B \quad \mid \exists(e_i \texttt{ for } l_i \texttt{ => } r_i) \in M,$
$l_i(A) > 0$ *and* $r_i(B) - l_i(B) < 0\}$
$\cup\{A$ *activates* $B \mid \exists(e_i \texttt{ for } l_i \texttt{ => } r_i) \in M,$
$l_i(A) > 0$ *and* $r_i(B) - l_i(B) > 0\}$

In particular, we have the following influences for elementary reactions of complexation, modification, synthesis and degradation:

$\alpha(\{\texttt{A + B => C}\}) = \{$ A inhibits B, A inhibits A, B inhibits A,
B inhibits B, A activates C, B activates C$\}$
$\alpha(\{\texttt{A = [C] => B}\}) = \{$ C inhibits A, A inhibits A, A activates B, C activates B$\}$
$\alpha(\{\texttt{A = [B] => \_}\}) = \{$ B inhibits A, A inhibits A$\}$
$\alpha(\{\texttt{\_ = [B] => A}\}) = \{$ B activates A$\}$

The inhibition loops on the reactants are justified by the negative sign in the Jacobian matrix of the differential semantics of such reactions. Unlike the differential influence graph, this graph is clearly trivial to compute by browsing the syntax of the rules:

Proposition 1. *The syntactical influence graph of a reaction model of n rules is computable in $O(n)$ time.*

3.3 Over-Approximation Theorem

Comparing the differential influence graph and the syntactical influence graph requires that the information in the kinetic expressions and in the reactions be compatible. This motivates the following definition where the first property forbids the presence of purely kinetic inhibitors not represented in the reaction, and the second property enforces that the variables appearing in the kinetic expressions do appear as reactants or enzymes in the reaction.

Definition 3. *In a reaction rule e* `for` *l=>r, we say that a kinetic expression e is* increasing *iff for all molecules x_k we have*

1. *$\partial e/\partial x_k \geq 0$ in all points of the space,*
2. *$l(x_k) > 0$ if $\partial e/\partial x_k > 0$ in some point of the space.*

A reaction model has an increasing kinetics *iff all its reaction rules have an increasing kinetics.*

One can easily check that:

Proposition 2. *Mass action law kinetics for any reaction, as well as Michaelis Menten and Hill kinetics for enzymatic reactions, are increasing.*

On the other hand, negative Hill kinetics of the form $k_1/(k_2^n + y^n)$ are not increasing. They represent an inhibition by a molecule y not belonging to the reactants, and thus not reflected in the syntax of the reaction.

Theorem 1. *For any reaction model with an increasing kinetics, the differential influence graph is a subgraph of the syntactical influence graph.*

Proof. If (A activates B) belongs to the differential influence graph then $\partial \dot{B}/\partial A > 0$. Hence there exists a term in the differential equation for B, of the form $(r_i(B) - l_i(B)) * e_i$ with $\partial e_i/\partial A$ of the same sign as $r_i(B) - l_i(B)$.

Let us suppose that $r_i(B) - l_i(B) > 0$ then $\partial e_i/\partial A > 0$ and since e_i is increasing we get that $l_i(A) > 0$ and thus that (A activates B) in the syntactical graph. If on the contrary $r_i(B) - l_i(B) < 0$ then $\partial e_i/\partial A < 0$, which is not possible for an increasing kinetics.

If (A inhibits B) is in the differential graph then $\partial \dot{B}/\partial A < 0$. Hence there exists a term in the differential semantics, of the form $(r_i(B) - l_i(B)) * e_i$ with $\partial e_i/\partial A$ of sign opposite to that of $r_i(B) - l_i(B)$.

Let us suppose that $r_i(B) - l_i(B) > 0$ then $\partial e_i/\partial A < 0$, which is not possible for an increasing kinetics. If on the contrary $r_i(B) - l_i(B) < 0$ then $\partial e_i/\partial A > 0$ and since e_i is increasing we get that $l_i(A) > 0$ and thus that (A activates B) is in the syntactical influence graph. □

Corollary 1. *For any reaction model with an increasing kinetics, the differential influence graph restricted to the phase space w.r.t. some initial conditions, is a subgraph of the syntactical influence graph.*

Proof. Restricting the points of the phase space to those points that are accessible from some initial states, restricts the number of edges in the differential influence graphs which thus remains a subgraph of the syntactical influence graph. □

It is worth noticing that even in the simple case of mass action law kinetics, the differential influence graph may be a strict subset of the syntactical influence graph. For instance let x be the following model :

$k_1 * A$ for $A => B$

$k_2 * A$ for $\_ = [A] => A$

In the syntactical influence graph, *A activates B, A activates A* and *A inhibits A*, however $\dot{A} = (k_2 - k_1) * A$, hence $\partial\dot{A}/\partial A$ can be made always positive or always negative or always null, resulting in the absence of respectively, *A inhibits A, A activates A* or both, in the differential influence graph.

3.4 Equivalence Theorem

Definition 4. *In a reaction rule e* `for` *l=>r, a kinetic expression e is* strongly increasing *iff for all molecules x_k we have*

1. *$\partial e/\partial x_k \geq 0$ in all points of the space,*
2. *$l(x_k) > 0$ if and only if there exists a point in the space s.t. $\partial e/\partial x_k > 0$*

A reaction model has a strongly increasing kinetics *iff all its reaction rules have a strongly increasing kinetics.*

Note that *strongly increasing* implies *increasing.*

Proposition 3. *Mass action law kinetics for any reaction, as well as Michaelis Menten and Hill kinetics for enzymatic reactions, are strongly increasing.*

Proof. For the case of Mass action law, the kinetics are of the form:

$$e_i = k_i * \prod_{l=1}^{m} x_l^{l_i(x_l)}$$

with $k_i > 0$ and $l_i(x_l) \geq 0$. We thus have $\partial e_i/\partial x_k = 0$ if $l_i(x_k) = 0$ and $\partial e_i/\partial x_k = k_i * l_i(x_k) * x_k^{l_i(x_k)-1} \prod_{l \neq k} x_l^{l_i(x_l)}$ otherwise, which clearly satisfies (1) and (2).

In the case of Hill kinetics (of which Michaelis Menten is a subcase), we have:

$$e_i = \frac{V_m * x_s^n}{K_m^n + x_s^n}$$

for the reaction $x_s + x_e => x_p + x_e$ and where $V_m = k_2 * x_e^{tot} = k_2 * (x_e + k_1 * x_e * x_s/(k_{-1} + k_2))$ from the steady state approximation. It is obvious that

$\partial e_i/\partial x_k = 0$ for all x_k other than x_s and x_e since they do not appear in e_i and one can easily check that with all the constants n, k_1, k_{-1}, k_2 strictly positive, both $\partial e_i/\partial x_e$ and $\partial e_i/\partial x_s$ are greater than 0 at some point in the space. □

Lemma 1. *Let M be a reaction model with a strongly increasing kinetics,*

If (A activates B) is an edge in the syntactical influence graph, and not (A inhibits B), then (A activates B) belongs to the differential influence graph.

If (A inhibits B) is an edge in the syntactical influence graph, and not (A activates B), then (A inhibits B) belongs to the differential influence graph.

Proof. Since $\partial \dot{B}/\partial A = \sum_{i=1}^{n}(r_i(B) - l_i(B)) * \partial e_i/\partial A$ and all e_i are increasing we get that $\partial \dot{B}/\partial A = \sum_{\{i \leq n | l_i(A) > 0\}}(r_i(B) - l_i(B)) * \partial e_i/\partial A$.

Now if (A activates B) is in the syntactical influence graph, but not (A inhibits B), then all rules such that $l_i(A) > 0$ verify $r_i(B) - l_i(B) \geq 0$ and there is at least one rule for which the inequality is strict. We thus get that $\partial \dot{B}/\partial A$ is a sum of positive numbers, amongst which one is such that $r_i(B) - l_i(B) > 0$ and $l_i(A) > 0$ which, since M is strongly increasing, implies that there exists a point in the space for which $\partial e_i/\partial A > 0$. Hence $\partial \dot{B}/\partial A > 0$ at that point, and (A activates B) is thus in the differential influence graph.

For inhibition the same reasoning applies with the opposite sign for the $r_i(B) - l_i(B)$ and thus for the partial derivative $\partial \dot{B}/\partial A$. □

This lemma establishes the following equivalence result:

Theorem 2. *In a reaction model with a strongly increasing kinetics and where no molecule is at the same time an activator and an inhibitor of the same target molecule, the differential and syntactical influence graphs coincide.*

This theorem shows that for standard kinetic expressions, the syntactical influences coincide with the differential influences based on the signs of the coefficients in the Jacobian matrix, when no molecule is at the same time an activator and an inhibitor of the same molecule. The theorem thus provides a linear time algorithm for computing the differential influences in these cases, simply by computing the syntactical influences. It shows also that the differential influence graph is independent of the kinetic expressions.

Corollary 2. *The differential influence graph of a reaction model of n rules with a strongly increasing kinetics is computable in time $O(n)$ if no molecule is at the same time an activator and an inhibitor.*

Corollary 3. *The differential influence graph of a reaction model is independent of the kinetic expressions as long as they are strongly increasing, if no molecule is at the same time an activator and an inhibitor.*

4 Application to Kohn's Map of the Mammalian Cell Cycle Control

Kohn's map of the mammalian cell cycle control [2] has been transcribed in BIOCHAM to serve as a large benchmarking example of approx. 500 species and 800 rules [11].

On Kohn's map, the computation of activation and inhibition influences takes less than one second CPU time (on a PC 1,7GHz) for the complete model, showing the efficiency of the syntactical inference algorithm. The influence graph is composed of 1231 activation edges and 1089 inhibition edges.

Furthermore in this large example no molecule is both an activator and an inhibitor of the same target molecule. Theorem 2 thus entails that the computed influence graph is equal to the differential graph that would be obtained in any kinetic model of Kohn's map for any standard kinetic expressions and for any (non zero) parameter values.

Since there is a lot of kinetic data missing for such a big model, the possibility to nevertheless obtain the exact influence graph without having to estimate parameters or even to choose precise kinetic expressions is quite remarkable, and justifies the use of purely qualitative models for the analysis of feedback circuits.

5 Application to the Signal Transduction MAPK "cascade"

Let us consider the MAPK signalling model of [12]. Figure 1 depicts the reaction graph as a bipartite graph with round boxes for molecules and rectangular boxes for rules. Figure 2 depicts the syntactical influence graph, where activation (resp. inhibition) is materialized by plain (resp. dashed) arrows.

This computed graph reveals the negative feedbacks that are somewhat hidden in a purely directional signalling cascade of reactions. Furthermore, as no molecule is at the same time an activator and an inhibitor of a same molecule, this graph is largely independent of the kinetics of the reactions, and would be identical to the differential influence graph for any standard kinetic expressions with any (non zero) kinetic parameter values.

These negative feedbacks, a necessary condition for oscillations [4,8,9], have been formally analyzed in [13] and interpreted as enzyme sequestration in complexes. Furthermore, oscillations in the MAPK cascade model have been shown in [18].

The influence graph also exhibits positive circuits. These are a necessary condition for multistationarity [4,7] that has been observed in the MAPK model, and experimentally in Xenopus oocytes [14]. Note that the absence of circuit in the (directional) reaction graph of MAPK was misinterpreted as a counterexample to Thomas' rule in [14] because of a confusion between both kinds of graphs.

6 Adding a Syntax for Antagonists in Reaction Rules

The over-approximation theorem 1 may suggest to provide a syntax for antagonists (i.e. inhibitors) in reaction rules, and generalize the result to this setting. Note that the mixing of mechanistic reaction models with non-mechanistic

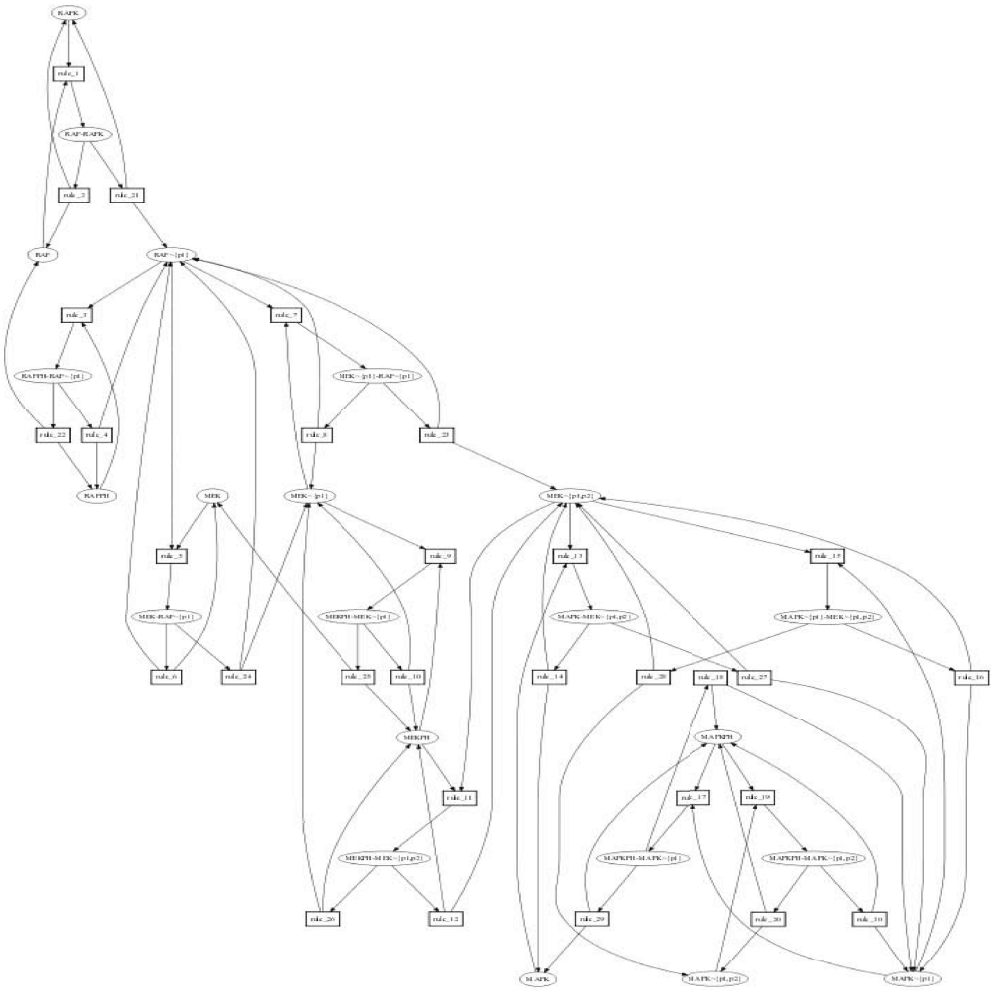


Fig. 1. Reaction graph of the MAPK model of[12]

information on the inhibitors of some reactions, is a common practice in diagrammatic notations which often combine reaction edges with activation and inhibition edges.

Let us denote by (e `for` l `=[/`a`]=>` r) a *generalized reaction rule* with antagonists a. Reaction rules with catalysts, of the form (e `for` l `=[c/`a`]=>` r), will remain an abbreviation for (e `for` $l + c$ `=[/`a`]=>` $r + c$). This notation for antagonists thus provides a counterpart for denoting the inhibitory effect of some agent on a reaction, symmetrically to the activation effect of the catalysts of the reaction.

Definition 5. *The* syntactical influence graph *associated to a generalized reaction model M is the graph having for vertices the molecular species, and for edges the following set:*

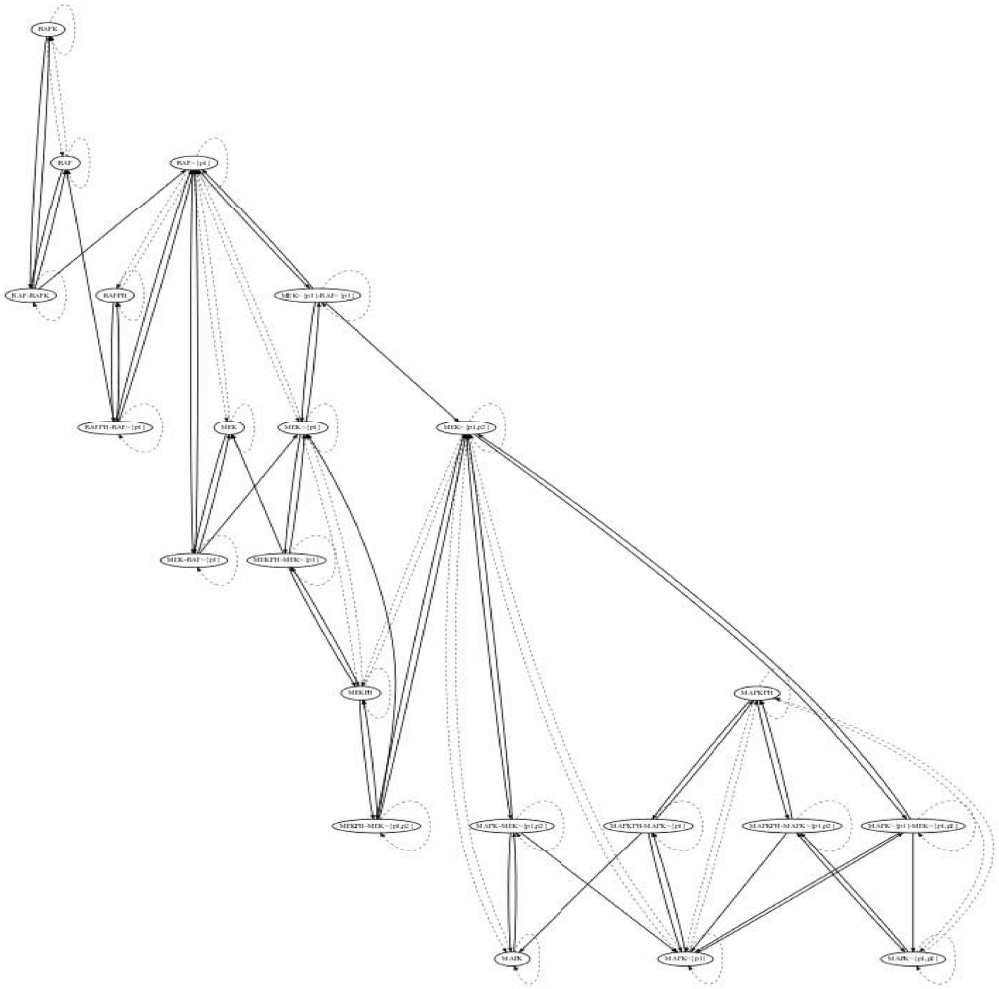


Fig. 2. Influence graph inferred from the MAPK reaction model

$$
\begin{array}{ll}
\{A \textit{ inhibits } B & \mid \exists (e_i \texttt{for } l_i \texttt{ =[/}a_i\texttt{]=> } r_i) \in M, \\
 & l_i(A) > 0 \textit{ and } r_i(B) - l_i(B) < 0\} \\
\cup\{A \textit{ activates } B & \mid \exists (e_i \texttt{for } l_i \texttt{ =[/}a_i\texttt{]=> } r_i) \in M, \\
 & l_i(A) > 0 \textit{ and } r_i(B) - l_i(B) > 0\} \\
\cup\{A \textit{ activates } B & \mid \exists (e_i \texttt{for } l_i \texttt{ =[/}a_i\texttt{]=> } r_i) \in M, \\
 & a_i(A) > 0 \textit{ and } r_i(B) - l_i(B) < 0\} \\
\cup\{A \textit{ inhibits } B & \mid \exists (e_i \texttt{for } l_i \texttt{ =[/}a_i\texttt{]=> } r_i) \in M, \\
 & a_i(A) > 0 \textit{ and } r_i(B) - l_i(B) > 0\}
\end{array}
$$

For instance, the set of syntactical influences of the generalized reaction rule `A =[/I]=> B`} is {*A inhibits A, I activates A, A activates B, I inhibits B*}. On the other hand, note that the definition of the differential influence graph applies to generalized reaction models as it is based on the kinetic expressions only.

Definition 6. *In a generalized reaction rule e* `for` l `=[/a]=>` r*, a kinetic expression e is* compatible *iff for all molecules x_k we have*

1. $l(x_k) > 0$ *if there exists a point in the space s.t.* $\partial e/\partial x_k > 0$,
2. $a(x_k) > 0$ *if there exists a point in the space s.t.* $\partial e/\partial x_k < 0$.

A generalized reaction model has a compatible kinetics *iff all its reaction rules have a compatible kinetics.*

For instance, a kinetics of the form `k1*Mdm2/(k2+P53)` for the generalized reaction rule `Mdm2 =[/P53]=> Mdm2p` expressing the phosphorylation of `Mdm2` that is inhibited by `P53` (see [19]) is compatible.

Note that for a reaction model, *strongly increasing* implies *compatible*. Furthermore, we have:

Theorem 3. *For any generalized reaction model with a compatible kinetics, the differential influence graph is a subgraph of the syntactical influence graph.*

Proof. If (A activates B) belongs to the differential influence graph then $\partial \dot{B}/\partial A > 0$. Hence there exists a term in the differential equation for B, of the form $(r_i(B) - l_i(B)) * e_i$ with $\partial e_i/\partial A$ of the same sign as $r_i(B) - l_i(B)$.

Let us suppose that $r_i(B) - l_i(B) > 0$ then $\partial e_i/\partial A > 0$, and since e_i is compatible we get that $l_i(A) > 0$ and thus that (A activates B) in the syntactical graph. If on the contrary $r_i(B) - l_i(B) < 0$ then $\partial e_i/\partial A < 0$, and since e_i is compatible we get that $a_i(A) > 0$ and thus that (A activates B) is in the syntactical influence graph.

If (A inhibits B) is in the differential graph then $\partial \dot{B}/\partial A < 0$. Hence there exists a term in the differential semantics, of the form $(r_i(B) - l_i(B)) * e_i$ with $\partial e_i/\partial A$ of sign opposite to that of $r_i(B) - l_i(B)$.

Let us suppose that $r_i(B) - l_i(B) > 0$ then $\partial e_i/\partial A < 0$, and since e_i is compatible we get that $a_i(A) > 0$ and thus that (A inhibits B) is in the syntactical influence graph. If on the contrary $r_i(B) - l_i(B) < 0$ then $\partial e_i/\partial A > 0$, and since e_i is compatible we get that $l_i(A) > 0$ and thus that (A activates B) is in the syntactical influence graph. □

This theorem shows that in this setting which mixes reaction rules with information on antagonists, the syntactical influence graph still over-approximates the differential influence graph for any standard kinetics.

7 Conclusion

This work shows that to a large extent, the influence graph of a reaction model is independent of the kinetic parameters and kinetic expressions, and that it can be computed in linear time simply from the syntax of the reactions. This happens for strongly increasing kinetics such as classical mass action law, Michaelis-Menten and Hill kinetics, when no molecule is at the same time an activator and an inhibitor of a same target molecule.

The inference of the syntactical influence graph from a reaction model has been implemented in BIOCHAM, and applied to various models. On a transcription of Kohn's map into approx. 800 reaction rules, this implementation shows that no molecule is at the same time an activator and an inhibitor of a same molecule, and therefore, our equivalence theorem states that the differential influence graph would be the same for any standard kinetics with any parameter values.

On the MAPK signalling cascade that does not contain any feedback reaction, the implementation does reveal both positive and negative feedback circuits in the influence graph, which has been a source of confusion for the correct application of Thomas' rules. Furthermore, in this example again, no molecule is at the same time an activator and an inhibitor of another molecule, showing the independence of the influence graph from the kinetics.

Acknowledgement. This work benefited from partial support of the European Union FP6 Network of Excellence REWERSE `http://www.rewerse.net`, and Strep TEMPO `http://www.chrono-tempo.org`.

References

1. Fages, F., Soliman, S.: Abstract interpretation and types for systems biology. Theoretical Computer Science (to appear, 2008)
2. Kohn, K.W.: Molecular interaction map of the mammalian cell cycle control and DNA repair systems. Molecular Biology of the Cell 10, 2703–2734 (1999)
3. Hucka, M., et al.: The systems biology markup language (SBML): A medium for representation and exchange of biochemical network models. Bioinformatics 19, 524–531 (2003)
4. Thomas, R., Gathoye, A.M., Lambert, L.: A complex control circuit: regulation of immunity in temperate bacteriophages. European Journal of Biochemistry 71, 211–227 (1976)
5. Kaufman, M., Soulé, C., Thomas, R.: A new necessary condition on interaction graphs for multistationarity. Journal of Theoretical Biology 248, 675–685 (2007)
6. Soulé, C.: Mathematical approaches to differentiation and gene regulation. C. R. Biologies 329, 13–20 (2006)
7. Soulé, C.: Graphic requirements for multistationarity. ComplexUs 1, 123–133 (2003)
8. Snoussi, E.: Necessary conditions for multistationarity and stable periodicity. J. Biol. Syst. 6, 3–9 (1998)
9. Gouzé, J.L.: Positive and negative circuits in dynamical systems. J. Biol. Syst. 6, 11–15 (1998)
10. Thomas, R.: On the relation between the logical structure of systems and their ability to generate multiple steady states or sustained oscillations. Springer Ser. Synergetics 9, 180–193 (1981)
11. Chabrier-Rivier, N., Chiaverini, M., Danos, V., Fages, F., Schächter, V.: Modeling and querying biochemical interaction networks. Theoretical Computer Science 325, 25–44 (2004)
12. Levchenko, A., Bruck, J., Sternberg, P.W.: Scaffold proteins biphasically affect the levels of mitogen-activated protein kinase signaling and reduce its threshold properties. PNAS 97, 5818–5823 (2000)

13. Ventura, A.C., Sepulchre, J.A., Merajver, S.D.: A hidden feedback in signaling cascades is revealed. PLoS Computational Biology (to appear, 2008)
14. Markevich, N.I., Hoek, J.B., Kholodenko, B.N.: Signaling switches and bistability arising from multisite phosphorylation in protein kinase cascades. Journal of Cell Biology 164, 353–359 (2005)
15. Calzone, L., Fages, F., Soliman, S.: BIOCHAM: An environment for modeling biological systems and formalizing experimental knowledge. BioInformatics 22, 1805–1807 (2006)
16. Fages, F., Soliman, S., Chabrier-Rivier, N.: Modelling and querying interaction networks in the biochemical abstract machine BIOCHAM. Journal of Biological Physics and Chemistry 4, 64–73 (2004)
17. Gillespie, D.T.: Exact stochastic simulation of coupled chemical reactions. Journal of Physical Chemistry 81, 2340–2361 (1977)
18. Qiao, L., Nachbar, R.B., Kevrekidis, I.G., Shvartsman, S.Y.: Bistability and oscillations in the huang-ferrell model of mapk signaling. PLoS Computational Biology 3, 1819–1826 (2007)
19. Ciliberto, A., Novák, B., Tyson, J.J.: Steady states and oscillations in the p53/mdm2 network. Cell Cycle 4, 488–493 (2005)

4.2 Relating ODE Systems and Reaction Models

The preceding hierarchy of semantics shows how to extract an ODE system from a reaction model, and there is little surprise in that. The reverse operation, necessary to apply all the structural methods described in this manuscript, and others, proves however to be much more subtle.

This time we do not fully agree with the answer given in the SBML FAQ about using MATLAB® as a model format:

> MATLAB is a fantastic system, but neither "MATLAB" or its scripting language is a model representation format.
>
> A MATLAB file is simply a script written in a proprietary application language. This is similarly true of Mathematica format, custom C++ programs, etc. So in some deep sense, these formats are not like SBML, which aims to describe the semantic components of the model.
>
> Equally important, the reality is that MATLAB and Mathematica scripts are really only fully runnable in only those applications. Some other systems such as Octave will run some MATLAB programs, but for many purposes, a person really has to go to MATLAB. That limits scientists' ability to reproduce each others' results, because using MATLAB or Mathematica ties them to commercial, closed-source systems. (SMBL FAQ – Isn't MATLAB a perfectly fine format for representing models?)

Actually, the most crucial point, missed by this answer though hinted at near the end of the second paragraph, is that contrary to SBML, an ODE system does not provide a clear structure for the underlying biochemical system. In other words, it simply does not define reactions.

4.2.1 Inferring Reaction Systems from ODEs

The first approach described here is a practical one, coming from our experience with BIOCHAM users coming from mathematical biology and looking for some way to get back the implicit structure in their ODEs. The demand for this is so great that the engineer in charge of BIOCHAM's web interface recently had to develop a specific web service for ODE to SBML conversion[2].

Actually, on top of very pragmatic and heuristic considerations, this article also presents precise mathematical conditions on what a system of ODEs coming from a biochemical reaction system should look like. This is in turn used to automatically curate some SBML models.

[5] François Fages, Steven Gay, and Sylvain Soliman. "Inferring Reaction Systems from Ordinary Differential Equations". In: *Theoretical Computer Science* 599 (Sept. 2015), pp. 64–78. ISSN: 0304-3975. DOI: 10.1016/j.tcs.2014.07.032

[2]The now much more complete API for this kind of service can be found at `http://lifeware.inria.fr/biocham/DOC/rest-api.html#model-conversion-model-export-post`

Doctopic: Theory of natural computing ARTICLE IN PRESS TCS:9840

Theoretical Computer Science ••• (••••) •••–•••

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Inferring reaction systems from ordinary differential equations ☆

François Fages *, Steven Gay *, Sylvain Soliman *

Inria Paris-Rocquencourt, France

ARTICLE INFO

Article history:
Received 31 January 2013
Received in revised form 25 June 2014
Accepted 28 July 2014
Available online xxxx

Keywords:
Systems biology
Chemical reaction network theory
Ordinary differential equations
SBML

ABSTRACT

In Mathematical Biology, many dynamical models of biochemical reaction systems are presented with Ordinary Differential Equations (ODE). Once kinetic parameter values are fixed, this simple mathematical formalism completely defines the dynamical behavior of a system of biochemical reactions and provides powerful tools for deterministic simulations, parameter sensitivity analysis, bifurcation analysis, etc. However, without requiring any information on the reaction kinetics and parameter values, various qualitative analyses can be performed using the *structure* of the reactions, provided the reactants, products and modifiers of each reaction are precisely defined. In order to apply these structural methods to parametric ODE models, we study a mathematical condition for expressing the consistency between the structure and the kinetics of a reaction, without restricting to Mass Action law kinetics. This condition, satisfied in particular by standard kinetic laws, entails a remarkable property of independence of the influence graph from the kinetics of the reactions. We derive from this study a heuristic algorithm which, given a system of ODEs as input, computes a system of reactions with the same ODE semantics, by inferring well-formed reactions whenever possible. We show how this strategy is capable of automatically curating the writing of ODE models in SBML, and present some statistics obtained on the model repository biomodels.net.

1. Introduction

In Mathematical Biology, many models are presented as a system of Ordinary Differential Equations (ODEs). Once the kinetic parameter values are fixed, this simple mathematical formalism completely defines the dynamical behavior of a system of biochemical reactions. It provides powerful tools for both transient and steady-state analysis via numerical integration, parameter sensitivity analysis, or bifurcation analysis, but only when kinetic information is available.

In absence of knowledge on the kinetics of each reaction, various qualitative analyses can nevertheless be performed using the *structure* of the reactions. This approach has rapidly developed in Systems Biology for reasoning on large interaction networks, with for instance, the analysis of qualitative attractors in a logical dynamics of gene networks *à la* Thomas [3–5], reachability and temporal logic properties in reaction networks [6–10], structural invariants in the Petri net representation of the reactions [11–16], or model reductions using graph theory concepts [17,18]. These qualitative analysis tools do not rely on kinetic information, but on the structure of the reaction network which has thus to be correctly written as a set of

☆ This paper is an extended version of a communication presented at CMSB'12. The algorithms described in this paper are implemented in the open-source software modeling platform Biocham [1,2] available at http://lifeware.inria.fr/biocham/ release 3.5. The models used in the experiments are available from http://www.biomodels.net/ release 24.

* Corresponding authors.

http://dx.doi.org/10.1016/j.tcs.2014.07.032
0304-3975/

Please cite this article in press as: F. Fages et al., Inferring reaction systems from ordinary differential equations, Theor. Comput. Sci. (2014), http://dx.doi.org/10.1016/j.tcs.2014.07.032

Doctopic: Theory of natural computing ARTICLE IN PRESS TCS:9840

formal reactions, with well-identified reactants, products and modifiers (and in certain cases their stoichiometry) for each reaction.

For instance, in [19], it is elaborated that structural information hidden in kinetic laws may affect the results obtained from structural analyses, such as elementary mode analysis [20], flux balance analysis [21], chemical organization theory [22], deficiency analysis or chemical reaction network theory [23,24].

It is worth noticing that these structural analyses may also directly support dynamic analyses. For instance, [25] applies network decomposition for a modular parameter estimation approach, [13] introduces a structural persistence criterion, Petri net place invariants reveal conservation laws in [26], while transition invariants can be used to identify fragile nodes and the core of a network [27], or to determine steady state solutions [28].

Furthermore, knowing the correct structure of each reaction is mandatory when a reaction network must be interpreted as a stochastic process (Continuous-Time Markov Chain, CTMC) *à la* Gillespie [29].

The question of the correct identification of a structured reaction model from a system of ODEs is thus important and is not new. Actually for the restricted case of models with only Mass Action kinetics a general solution is provided in [30]. This approach was evolved over the years, see for instance [31] for sparse/dense/core solutions when numerical values are provided for the parameters, or [32] for unicity conditions in the symbolic case, still in the restricted framework of mass action law kinetics. In [19], the authors present an algorithm that uncovers hidden structural information for some Systems Biology Markup Language (SBML) [33,34] models of the biomodels.net repository [35], with restricting to reaction models without inhibitors.

In this paper, we describe an algorithm for finding a reaction models for a given system of ODEs, considering reaction with inhibitors and general kinetic expressions. The first contribution of this paper is to propose a mathematical condition for expressing the consistency between the kinetic expression and the reactant-product-inhibitor structure of a reaction. We introduce well-formedness (Definition 2.5) and strictness (Definition 2.6) conditions for reactions, and show that they are satisfied by standard kinetics such as Mass Action law, Michaelis–Menten, Hill and negative Hill kinetics. The well-formedness condition is also shown to entail a property of independence of the influence graph (or symbolic Jacobian matrix) from the kinetics of the reactions (Theorem 2.16). This result generalizes a previous result in [36,37] to reactions with inhibitors. It shows that the influence graph of a well-formed reaction system with inhibitors is essentially independent of the kinetics, can be computed in linear time in the number of reactions when the number of species per reaction is bounded, and can thus advantageously be used to perform multi-stationarity analyses by circuit analysis *à la* Thomas [38–43,5,4,3].

The second contribution of this paper is to use these well-formedness and strictness conditions to prove the completeness of a new general algorithm for inferring a reaction system equivalent to an ODE system. This algorithm, of time complexity in $O(n \times t)$ where n is the number of variables and t the number of terms in the ODE, is shown to preserve the ODE semantics of the reactions (soundness 3.10), as well as their well-formedness when applied to the ODE semantics of a non-decomposable well-formed reaction system (weak completeness 3.12).

Our third contribution is to show that our algorithm can be used to automatically curate the writing of ODE models with reactions, as required in SBML. The fact that SBML has become a standard for sharing and publishing models has helped in making modelers formalize the reaction structure of their models. Unfortunately, SBML does not enforce any strong coherence between the structure and the kinetics of a reaction. Therefore the structural interpretation of models transcribed in reaction-based formalisms such as SBML may vary according to different choices of representation of the original ODE model as a reaction system, and may invalidate some structural analyses. We compare our results to the one presented in [19], and provide some statistics obtained on the rewriting in SBML of the curated part of the biomodels.net repository, showing that our method is able to automatically decrease the number of non-well-formed reaction systems from 65% to 29%.

2. A theory of well-formed reactions and kinetics

In this section, we consider a finite set $S = \{x_1, \ldots, x_s\}$ of s molecular species, and a finite set of n reactions over S which are formally represented as multiset rewriting rules with kinetic expressions.

Multisets are used for representing reactants, products and inhibitors in reactions. A multiset s of molecular species is a function $S \longrightarrow \mathbb{N}$ which gives the number (stoichiometric coefficient) $s(x)$ of each molecular species $x \in S$ in s. We have $s(x) = 0$ if x does not belong to s, and $s(x) \geq 1$ if x belongs to s, which is also written $x \in s$ by abuse of notation. The empty multiset is written $\emptyset$. Equivalently, a multiset s will also be denoted by the linear expression $\sum_{i=1}^{m} s(x_i) \times x_i$, which gives the stoichiometric coefficients of each molecular species x_i in s. This corresponds to the classical chemical notation $2\mathrm{H} + \mathrm{O} \longrightarrow \mathrm{H_2O}$.

We shall now introduce the well-formedness and strictness conditions and describe some of their properties.

2.1. Well-formedness and strictness conditions

In the following definition, a reaction is composed of multisets for reactants, products and inhibitors that are not assumed to be disjoint.

Please cite this article in press as: F. Fages et al., Inferring reaction systems from ordinary differential equations, Theor. Comput. Sci. (2014), http://dx.doi.org/10.1016/j.tcs.2014.07.032

Doctopic: Theory of natural computing ARTICLE IN PRESS TCS:9840

Definition 2.1. A *reaction* is a quadruple (r, m, p, f), where r is the multiset of *reactants*, m the multiset of *inhibitors*, p the multiset of *products*, and f, called *kinetic expression*, a mathematical function over molecular species concentrations, $f : \mathbb{R}^S \longrightarrow \mathbb{R}$. A *reaction system* is a finite set of reactions.

The species that are both reactants and products in a reaction are called *catalysts*. For the sake of readability, a reaction (r, m, p, f) will also be written $r/ \xrightarrow{f} p$ or just $r \xrightarrow{f} p$ if it has no inhibitor, i.e. when $m = \emptyset$. The kinetic expression will also be omitted if it is not relevant.

Example 2.2. For instance, the following reaction, transcribed from Kohn's map of the cell cycle [44],

$$pMPF + Cdc25 \xrightarrow{k_1 \times pMPF \times Cdc25} MPF + Cdc25$$

expresses the activation of the Mitosis Promoting Factor *MPF* by the kinase *Cdc*25. It has as rate law $f = k_1 \times pMPF \times Cdc25$, i.e. a Mass Action kinetics with parameter k_1. In this reaction, *pMPF* is a reactant, *MPF* a product, *Cdc*25 a reactant and a product at the same time, i.e. a catalyst in our terminology, and there is no inhibitor.

A simplified version of that reaction can be written by omitting the kinase *Cdc*25, as follows:

$$pMPF \xrightarrow{V \times pMPF/(K+pMPF)} MPF$$

That form typically derives from three reactions describing the reversible association of *pMPF* and *Cdc*25 and the dissociation to *MPF*, by making a quasi steady state approximation on *Cdc*25, which results in a Michaelis–Menten kinetics with parameters K and V.

It is worth noting that in a reaction, a reactant or a product can also be an inhibitor if it appears in m.

Example 2.3. For instance, the Botts–Morales general modifier mechanism accounts for a modifier M that can enhance and slow down a reaction $A \longrightarrow B$, depending on its concentration [45]. This can be represented in our setting by a reaction of the form

$$A + M/M \longrightarrow B + M.$$

SBML does not distinguish between catalysts and inhibitors which are just considered as "modifiers" in SBML annotations. However we find it useful for the theory developed here to distinguish between the activation or inhibitory effects of a modifier, and mark it syntactically as such in the structure of the reaction. If a modifier has both activation and inhibitory effects, it will just appear in r, m and p in our setting, without loss of generality.

It is also worth noting that we consider only irreversible reactions, as in Feinberg's Chemical Reaction Network theory [23]. A reversible reaction is thus represented by two reactions, one for each direction. This is one important difference with the Systems Biology Markup Language (SBML) that permits the declaration of a reversible reaction with only one single kinetic expression which can be negative.

These distinctions do not affect the system of ODEs that is classically associated to a reaction system by the Reaction Rate Equation as follows:

Definition 2.4. The ODE semantics of a reaction system

$$R = \{r_i/m_i \xrightarrow{f_i} p_i\}_{i=1,\ldots,n}$$

over molecules $\{x_1, \ldots, x_s\}$, is the system of ordinary differential equations

$$\dot{x}_j = \sum_{i=1}^{n} \big(p_i(x_j) - r_i(x_j)\big) \times f_i$$

for $1 \leq j \leq s$.

Our aim is to go in the reverse direction, that is to infer from any ODE system a reaction system with the same ODE semantics. Let us first remark that any ODE system $\dot{x}_j = g_j$ can be trivially transcribed in a reaction system using artificial synthesis reactions for each molecular species, with the terms of the differential equation as kinetic expressions, as follows:

$$\emptyset \xrightarrow{g_j} x_j$$

Since the ODE semantics is identical to the original ODE system, this is correct as far as numerical simulations are concerned, but prevents the use of structural analysis methods or stochastic simulations as the structures of the reactions are totally

Please cite this article in press as: F. Fages et al., Inferring reaction systems from ordinary differential equations, Theor. Comput. Sci. (2014), http://dx.doi.org/10.1016/j.tcs.2014.07.032

Doctopic: Theory of natural computing ARTICLE IN PRESS TCS:9840

meaningless. It is worth remarking that some ODE models have nevertheless been transcribed in SBML using that scheme, since it does not affect simulations. This is the case for instance of model `BIOMD0000000008.xml` in biomodels.net for the ODE model of [46]. We will use that example in Section 3.1 to illustrate our reaction inference algorithm and its capability of automatically curating the writing in SBML of ODE models as reaction systems.

In order to try to infer meaningful reactions from ODEs, we are interested in mathematical conditions for expressing the consistency of the kinetic expression f with the structure (r, m, p) of a reaction. Furthermore, since it is common practice to aggregate a system of elementary reactions in one abstract reaction with more complex kinetics (the simplest example of which are Michaelis–Menten and Hill kinetics for enzymatic reactions), we do not content ourselves with elementary kinetic expressions such as mass action law kinetics, but seek abstract consistency properties that can be applied to any mathematical expression given as kinetics. This is in contrast to most work on chemical reaction network theory [23,24,32], but in accordance with the use in SBML of MathML for writing the kinetic expressions without any limitation on the use of mathematical symbols.

Definition 2.5. A reaction (r, m, p, f) over molecular species $\{x_1, \ldots, x_s\}$ is *well-formed* if the following conditions hold:

1. $f(x_1, \ldots, x_s)$ is a partially differentiable function, non-negative on $\mathbb{R}^s_+$;
2. $x_i \in r$ if and only if $\partial f/\partial x_i(\vec{x}) > 0$ for some value $\vec{x} \in \mathbb{R}^s_+$;
3. $x_i \in m$ if and only if $\partial f/\partial x_i(\vec{x}) < 0$ for some value $\vec{x} \in \mathbb{R}^s_+$.

The first condition expresses that the kinetic expression must be a differentiable and non-negative function for all non-negative values of the variables. The second (resp. third) condition states that the partial derivative of f w.r.t. a reactant (resp. an inhibitor) must be positive (resp. negative) for some (not necessarily all) non-negative values of the variables.

It is worth noting that we do not impose the *monotonicity* condition that for any variable $x_i \in V$, $\partial f/\partial x_i$ should be either non-negative on the positive orthant, or non-positive on the positive orthant. In our setting, a molecular species can thus be both a reactant and an inhibitor in a well-formed reaction, depending on the values of the concentrations. On the other hand we shall make use of the following:

Definition 2.6. A reaction (r, m, p, f) is *strict* if its kinetics $f(x_1, \ldots, x_s) = 0$ whenever $x_j = 0$ for any x_j such that $r(x_j) > 0$.

This condition expresses that the kinetics must be zero if the concentration of one of the reactants is zero. If the kinetics is a rational expression, that strictness condition implies that the kinetic expression is a product of the reactants with a fractional expression defined for all non-negative values of the variables. More generally it enforces the positivity of the system:

Definition 2.7 *(Positive System).* A dynamical system over $\mathbb{R}^k$ is called positive if $\mathbb{R}^k_+$ is an invariant set for the system, i.e., $\forall x_0 \geq 0$, $t \geq 0$, $x(t, x_0) \geq 0$.

Proposition 2.8 *(Positivity). The ODE semantics of a well-formed and strict reaction system defines a positive system.*

Proof. In Definition 2.4 we have $\dot{x}_j = \sum_{i=1}^{n}(p_i(x_j) - r_i(x_j)) \times f_i$ and since the system is well-formed, the f_i are all non-negative. The only negative terms thus have $r_i(x_j) > 0$ and from the strictness condition this entails that $f_i = 0$ when $x_j = 0$. Hence $\dot{x}_j \geq 0$ whenever $x_j = 0$ since it is a sum of non-negative terms. Therefore x_j cannot become negative when its initial value is non-negative, and since this holds for all j, the system is positive. □

The strictness condition excludes the writing of a reversible reaction with one single reaction by summing the kinetic expressions of each direction, as allowed in SBML, when the reactants differ from the products.

It also excludes the existence of a strict well-formed reaction system for any ODE system, as shown by

Example 2.9. The equation $\dot{x} = -k$ is not the ODE semantics of any strict well-formed reaction system, since that ODE defines a non-positive system (Proposition 2.8). That ODE can be associated to the non-strict well-formed reaction system

$$x \xrightarrow{l \times x} 2 \times x$$
$$x \xrightarrow{k+l \times x} \emptyset$$

(where the kinetic expression is not null when $x = 0$). This is the result computed in that case by Algorithm 3.6 described later.

Please cite this article in press as: F. Fages et al., Inferring reaction systems from ordinary differential equations, Theor. Comput. Sci. (2014), http://dx.doi.org/10.1016/j.tcs.2014.07.032

Doctopic: Theory of natural computing ARTICLE IN PRESS TCS:9840

Example 2.10. As for an example with inhibitors, let us consider the following three reactions representing the core of the action of the Circadian clock on the Cell Cycle, as described by Matsuo et al. [47]:

$$pMPF + Cdc25 \xrightarrow{k_1 \times pMPF \times Cdc25} MPF + Cdc25$$
$$MPF + Wee1 \xrightarrow{k_2 \times MPF \times Wee1} pMPF + Wee1$$
$$\emptyset / Clock \xrightarrow{k_3/(k_4+Clock)} Wee1$$

where k_1, k_2, k_3 are parameters. The first reaction is the one of Example 2.2. Those reactions are well-formed and strict. In particular, we have $\partial \dot{Wee1}/\partial Clock = \partial(k_3/(k_4 + Clock))/\partial Clock = -k_3/(k_4 + Clock)^2 < 0$ for showing the inhibitory effect of Clock in the synthesis reaction of Wee1. Their ODE semantics is

$$\dot{pMPF} = k_2 \times MPF \times Wee1 - k_1 \times pMPF \times Cdc25$$
$$\dot{MPF} = k_1 \times pMPF \times Cdc25 - k_2 \times MPF \times Wee1$$
$$\dot{Wee1} = k_3/(k_4 + Clock)$$
$$\dot{Cdc25} = 0$$
$$\dot{Clock} = 0$$

The well-formedness and strictness conditions are satisfied by standard kinetic laws. One can easily check

Proposition 2.11. *Reactions with mass action law kinetics:*

$$\sum_j n_j \times x_j \xrightarrow{k \times \prod_j x_j^{n_j}} p$$

Michaelis–Menten kinetics:

$$x \xrightarrow{V \times x/(K+x)} y$$

Hill kinetics:

$$x \xrightarrow{V \times x^n/(K^n+x^n)} y$$

or negative Hill kinetics:

$$\emptyset / x \xrightarrow{V/K^n+x^n} y$$

with rate constants $k, V, K > 0$ and exponent $n \geq 1$, are well-formed and strict.

We shall see in Section 4.2 that these conditions are currently violated in a majority of reaction systems of the biomodels.net repository, but that most of them can be automatically corrected by modifying their structure and writing in SBML, without changing their ODE semantics.

2.2. Influence graph associated to a well-formed reaction system

The influence graph between molecular species induced by the ODE semantics of a well-formed reaction system enjoys a remarkable property of independence from the kinetics, which we present in this section. Influence graphs have been initially introduced in the setting of gene regulatory networks [3] as a simple abstraction enabling reasoning about complex regulation mechanisms. These graphs completely abstract from the precise interactions, especially at post-transcriptional level, and retain only the activation and inhibitory effects on gene transcription. As conjectured in [4], the existence of a positive circuit (resp. a negative circuit) in an influence graph has been proved to be a necessary condition for multi-stationarity, e.g. for cell differentiation, (resp. for oscillations, e.g. for homeostasis) in different formalisms, and in particular for ODE systems in [43,39–42] and recently in [38] for the ODE semantics of non-linear reaction systems.

Here, we show that in a well-formed reaction system, and under a very general assumption, the influence graph of the reactions is identical to the influence graph of the ODE semantics of the reactions.

On the one hand, in an ODE system, the influence graph is mathematically defined by the signs of the coefficients in the Jacobian matrix of the system, $(\partial \dot{x}_i/\partial x_j)$, as follows:

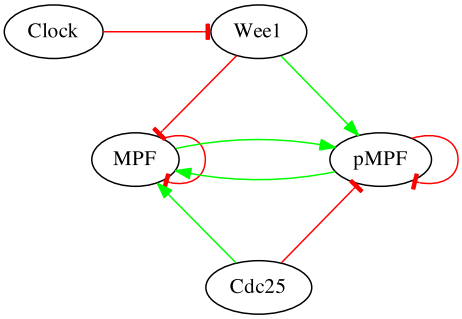
Please cite this article in press as: F. Fages et al., Inferring reaction systems from ordinary differential equations, Theor. Comput. Sci. (2014), http://dx.doi.org/10.1016/j.tcs.2014.07.032

Doctopic: Theory of natural computing ARTICLE IN PRESS TCS:9840

Definition 2.12. The *differential influence graph (DIG)* associated to (the ODE semantics of) a reaction system is the graph that has for vertices the molecular species, and for labeled edges the following set of signed edges:

$$\{x_i \longrightarrow^+ x_j \mid \partial \dot{x}_j/\partial x_i(\vec{x}) > 0 \text{ for some value } \vec{x} \in \mathbb{R}^s_+\}$$
$$\cup \{x_i \longrightarrow^- x_j \mid \partial \dot{x}_j/\partial x_i(\vec{x}) < 0 \text{ for some value } \vec{x} \in \mathbb{R}^s_+\}$$

Example 2.13. The DIG of Example 2.10 can be depicted by the following graph:



where the positive influences are represented by gray (green in web version) arrows with triangular tips, and negative influences are represented by black (red in web version) arrows with blunt tips. For instance, the negative influence of Clock on Wee1 comes from the negative sign of $\partial \dot{Wee1}/\partial Clock$ as detailed in Example 2.10. There are negative loops on *MPF* and *MPF* since $\partial \dot{MPF}/\partial MPF = -k_2 Wee1 < 0$ and $\partial p\dot{MPF}/\partial pMPF = -k_2 Cdc25 < 0$, and not on $Cdc25$ since $\dot{Cdc25} = 0$. Note that a useful circuit analysis in this example would necessitate considering the reactions of formation of MPF and is beyond the scope of this paper.

On the other hand, in a reaction system, one can define an influence graph directly from the stoichiometry of the reactions, ignoring the kinetics, as follows:

Definition 2.14. The *stoichiometric influence graph (SIG)* associated to a finite set R of reactions is the graph that has for vertices the molecular species, and for labeled edges the following set of signed edges:

$$\{x \longrightarrow^+ y \mid \text{either } p_i(y) - r_i(y) > 0 \text{ and } r_i(x) > 0, \text{ or } p_i(y) - r_i(y) < 0 \text{ and } m_i(x) > 0, \text{ for some reaction } i\}$$
$$\cup \{x \longrightarrow^- y \mid \text{either } p_i(y) - r_i(y) < 0 \text{ and } r_i(x) > 0, \text{ or } p_i(y) - r_i(y) > 0 \text{ and } m_i(x) > 0, \text{ for some reaction } i\}$$

Intuitively, there is a positive (resp. negative) arc from x to y if x is a reactant in a reaction that produces more (resp. less) y than it consumes, or an inhibitor in a reaction that consumes more (resp. less) y than it produces.

Unlike the DIG, which needs to compute the sign of partial derivatives, the SIG can be easily computed in linear time in the number of reactions, assuming that the number of species per reaction is bounded, since it is sufficient to parse the stoichiometric coefficients of the reactions. As already shown in [36], the SIG is an over-approximation of the DIG:

Theorem 2.15. *(See [36].) For any finite set R of well-formed reactions, the DIG of R is a subgraph of the SIG of R.*

We show here that, even in the presence of inhibitors, the SIG is in fact identical to the DIG with an extra assumption. Let us say that a tuple of molecular species (x, y) is in *conflict in an influence graph* if we have both $x \longrightarrow^+ y$ and $x \longrightarrow^- y$.

Theorem 2.16. *For any finite set R of well-formed reactions such that the SIG of R contains no conflict, the DIG and the SIG are identical.*

Proof. We just have to prove that the SIG is a subgraph of the DIG. Let us consider an arc $x \longrightarrow^+ y$ in the SIG. By Definition 2.14 there exists a reaction i with either $p_i(y) - r_i(y) > 0$ and $r_i(x) > 0$, or $p_i(y) - r_i(y) < 0$ and $m_i(x) > 0$. Since the reaction is well-formed, we have either $p_i(y) - r_i(y) > 0$ and $\partial f_i/\partial x(\vec{z}) > 0$, or $p_i(y) - r_i(y) < 0$ and $\partial f_i/\partial x(\vec{z}) < 0$, for some $\vec{z} \in \mathbb{R}^s_+$. Now, if $p_i(y) - r_i(y) > 0$ then f_i occurs in $\dot{y}$ with a positive sign. Since $\partial f_i/\partial x(\vec{z}) > 0$ and there is no conflict in the SIG, we thus get $\partial \dot{y}/\partial x(\vec{z}) > 0$, i.e. $x \longrightarrow^+ y$ is in the DIG. Similarly, if $p_i(y) - r_i(y) < 0$, f_i occurs in $\dot{y}$ with a negative sign and $\partial f_i/\partial x(\vec{z}) < 0$, hence $\partial \dot{y}/\partial x(\vec{z}) > 0$, i.e. $x \longrightarrow^+ y$ is in the DIG. The proof for an arc $x \longrightarrow^- y$ in the SIG is symmetrical. □

Corollary 2.17. *The DIG of a finite set of well-formed reactions without conflict in its SIG, is independent of the kinetic expressions.*

Please cite this article in press as: F. Fages et al., Inferring reaction systems from ordinary differential equations, Theor. Comput. Sci. (2014), http://dx.doi.org/10.1016/j.tcs.2014.07.032

Corollary 2.18. *The DIG of a finite set of well-formed reactions without conflict in its SIG, is computable in linear time in the number of reactions, when the number of species appearing in a reaction is bounded.*

The SIG of Example 2.10 is trivial to compute and since it contains no conflict, we can predict by Theorem 2.16 that it is identical to its DIG depicted in Example 2.13.

Example 2.19. As for an example of conflict, in the simplified model of the yeast cell cycle of [48], the double activation reactions of MPF through Cdc25 and Wee1 [44], are simplified in a single autocatalytic reaction in parallel with a deactivation reaction:

$$pMPF + MPF \longrightarrow 2 \times MPF$$
$$MPF \longrightarrow pMPF$$

Such reactions create a conflict in the SIG, namely $MPF \longrightarrow^{-} pMPF$ and $MPF \longrightarrow^{+} pMPF$. In general, there is a possibility that such conflicting direct influences in the SIG may be balanced in the ODEs and do not appear in the DIG. This situation is however quite pathological and rare in practice, and occurs when over-simplifications are made. For instance, Kohn's map of the cell cycle control [44] contains 800 reactions [8] and does not contain any conflict in its SIG [36]. The conflict of influences between MPF and pMPF in Tyson's model comes from the compression in one loop of the two positive circuits through Wee1 and Cdc25 respectively. The decompression of this loop makes disappear the influence conflict.

Thomas's necessary condition for a system to exhibit multi-stationarity is the existence of a positive circuit, i.e., a simple oriented cycle such that the product of the signs of its edges is positive, in the DIG [40]. That condition has proven useful to reason about gene interaction networks and predict the possibilities of multi-stationarity, i.e. cell differentiation. However, Thomas's original condition provides no information in presence of reactions with two reactants, since a reaction like for instance $A + B \longrightarrow C$ immediately creates a positive circuit of negative influences between A and B in the associated SIG and DIG for any reasonable kinetics. This counter-example has been recently rule out in [38], where it is shown that Thomas's conditions can be made stronger for reactions models, by labeling the influence edges by the reactions they come from, and by restricting the analysis of circuits to circuits labeled by different reactions. With this stronger condition for multi-stationarity, the analysis of labeled circuits in the DIG of a reaction system does provide information on its capabilities of exhibiting multi-stationarity. Theorem 2.16 shows, perhaps surprisingly, that for well-formed reaction systems without conflicts, the DIG is essentially independent from the kinetics, and in fact identical to the SIG, which is easy to compute and can be used to perform multi-stationarity analysis by circuit analysis.

3. Reaction system inference algorithm

In this section we present an algorithm to infer a reaction system from an arbitrary ODE system, and study its properties. The algorithm proceeds in two steps: one first step for inferring hidden molecules corresponding to linear invariants of the ODE system, and one second step for inferring the reactions.

3.1. Motivating example

As remarked in Section 2.1, any ODE model can be transcribed in a reaction system using artificial synthesis and degradation reactions for each molecular species, with the positive, respectively negative, terms of the differential equation for the variables as kinetic expressions. While preserving the ODE semantics and thus ODE simulations, such a transcription prevents the use of structural methods and stochastic simulations to analyze the system.

Such a transcription has nevertheless been used in biomodels.net to write the ODE model of [46] in SBML and create `BIOMD0000000008.xml`. This model adds a control mechanism to the cell-cycle model of Goldbeter et al. in [49] but with this transcription in SBML, the reaction graph is not even connected.

Here are some of the reactions of this model (after expansion of the macros used in the original writing) which illustrate the problem:

$$\emptyset \xrightarrow{(1-M)\times C\times V_1'\times(C+K_6^{-1})/(K_1+1-M)} M$$
$$M \xrightarrow{M\times V_2/(K_2+M)} \emptyset$$
$$\emptyset \xrightarrow{M\times V_3'\times(1-X)/(K_3+1-X)} X$$
$$X \xrightarrow{V_4\times X/(K_4+X)} \emptyset$$

One can notice that $\partial f_1/\partial C \neq 0$, where f_1 is the kinetic expression of the first reaction, but C is not a reactant nor an inhibitor. The model is therefore not well-formed.

Please cite this article in press as: F. Fages et al., Inferring reaction systems from ordinary differential equations, Theor. Comput. Sci. (2014), http://dx.doi.org/10.1016/j.tcs.2014.07.032

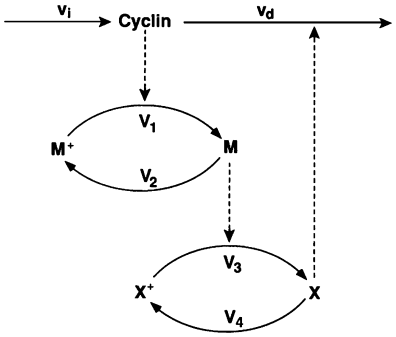
Doctopic: Theory of natural computing ARTICLE IN PRESS TCS:9840

One can also note that, though encoded in complicated MathML expressions, $1-M$ (resp. $1-X$) appears in the synthesis of M (resp. X) as a way to represent the inactive form of M (resp. X). Indeed, [49] states that *"$(1-M)$ thus represents the fraction of inactive (i.e., phosphorylated) cdc2 kinase, while $(1-X)$ represents the fraction of inactive (i.e., dephosphorylated) cyclin protease"*.

When applied to the ODE system associated to this model, the reaction system inference algorithm presented in the next two sections, infers two *hidden molecules* and the following well-formed and strict reactions:

$$M_i + C \xrightarrow{C\times V_1'\times(C+1/K_6)\times M_i/(K_1+M_i)} M + C$$
$$M \xrightarrow{V_2\times M/(K_2+M)} M_i$$
$$X_i + M \xrightarrow{M\times V_3'\times X_i/(K_3+X_i)} X + M$$
$$X \xrightarrow{V_4\times X/(K_4+X)} X_i$$

The two inactive forms are now explicitly represented by two inferred molecules, written M_i and X_i, and the actions of C on M and of M on X are properly transcribed. The reaction system inferred automatically from the ODE semantics is well-formed and strict, and in fact consistent with the graphical representation of the paper [49] where dashed arrows represent catalytic effects:



In that form, the inferred model is thus suitable for further structural analysis.

The following sections present the reaction system inference algorithm in two steps: first the algorithm for inferring hidden molecules corresponding, as above, to invariants, second the algorithm for inferring well-formed reactions whenever possible.

3.2. Inference algorithm for hidden molecules

ODE models often contain algebraic invariants, i.e., algebraic equations relating variables of the model and that hold true in any solution of the ODE system. Among those, linear invariants $\sum \lambda_i x_i = \Lambda$, e.g. mass conservation invariants, or Petri-net place invariants, are an important particular case. A linear invariant can be used to simplify a model by eliminating one variable and replacing it with a linear expression. This may have several advantages, but when writing the model with reactions, such simplifications performed on the ODE system need be reversed in order to restore the correct structure of the reactions on eliminated molecular species, as shown for instance in the previous section with the inactive forms M_i of M, and X_i of X.

A preprocessor is first applied before the reaction inference algorithm, in order to reverse the elimination of linear invariants and infer hidden molecules. The expressions f for which new molecules are introduced need be chosen with care in order to avoid the introduction of useless variables. Restricting the search to expressions of the form $k-x$ or $k-x-y$ where k is a constant or parameter, and x and y are molecule concentrations, has proven useful in practice. This leads to

Algorithm 3.1 *(Hidden molecule inference).*

input: ODE system O over variables $\{x_1,\ldots,x_s\}$,

1. iteratively replace in O any expression of the form $-x+y$ by $y-x$,
2. for each expression of the form $k-x-y$ in O where k is a numerical constant or a parameter, and x and y are variables,
   (a) introduce a new variable z with time derivative $\dot{z}=-\dot{x}-\dot{y}$, and functional dependency equation $z=k-x-y$,
   (b) substitute any occurrence of $k-x-y$ in O by z,
   (c) substitute any occurrence of $k+v-x-y$ in O for any expression v, by $v+z$,
   (d) substitute any occurrence of $k-x+w-y$ in O for any w, by $v+z$,

Please cite this article in press as: F. Fages et al., Inferring reaction systems from ordinary differential equations, Theor. Comput. Sci. (2014), http://dx.doi.org/10.1016/j.tcs.2014.07.032

3. for each expression $k - x$ appearing in O where k is a constant or a parameter and x a variable,
   (a) introduce a new variable z with time derivative $\dot{z} = -\dot{x}$ and functional dependency equation $z = k - x$,
   (b) substitute any occurrence of $k - x$ in O by z,
   (c) substitute any occurrence of $k + v - x$ in O for any expression v, by $z + v$,

output: ODE system O over variables $\{x_1, \ldots, x_s\}$ and hidden molecule variables $\{z_1, \ldots, z_k\}$, together with functional dependency equations $z_j = f_j(x_1, \ldots, x_s)$ for $1 \leq j \leq k$.

Proposition 3.2 *(Soundness). Let O be an ODE system over variables $\{x_1, \ldots, x_s\}$. The ODEs computed by Algorithm 3.1 for the time derivatives of $x_1, \ldots, x_s$, are mathematically equivalent to the equations in O given the functional dependency equations $z_j = f_j(x_1, \ldots, x_s)$ for the hidden molecules.*

Proof. We prove that each step of the algorithm replaces equal by equal, and thus that the whole execution preserves the mathematical equivalence of the equations. First, step 1 is a purely syntactical transformation that does not change the ODE system O. Now note that all the other changes are of two forms. Either the introduction of a new variable z such that $\dot{z} = \sum \lambda_i \dot{x}_i$, together with the functional dependency equation $z = k + \sum \lambda_i x_i$, steps 2(a) and 3(a). Since the differential equation on z is indeed the time derivative of the definition of z, this does not change the equations on $\dot{x}_i$. Or the replacement of $k + \sum \lambda_i x_i$ by z, steps 2(b–d) and 3(b–c), which are equal from the previous definition of z. □

3.3. Inference algorithm for reactions

The inference algorithm for reactions is based on a syntactical normal form for ODE systems which facilitates the recognition of common subterms in the equations.

We consider ODEs and kinetic laws written in MathML as terms with mathematical operations and functions (e.g. $+$, $-$, $/$, $\times$, etc.), constants of $\mathbb{R}$ and variables representing species concentrations and parameters. It is beyond the scope of this paper to precisely describe the mathematical expressions allowed and the symbolic computation performed. However, let us call *non-decomposable* a term that:

- its functor (top function symbol) is neither $+$ nor $-$;
- cannot be reduced at top-level by the algebraic laws of distributivity of the product and division on addition and subtraction, e.g. if its functor is $\times$ (resp. $/$) then the arguments (resp. the numerator) are not sums.

Definition 3.3. A reaction $r/m \xrightarrow{f} p$ over molecular species $\{x_1, \ldots, x_s\}$ is *non-decomposable* if f is syntactically a non-decomposable term.

Definition 3.4. A mathematical expression is in *additive normal form* if it is of the form $\sum_{i=1}^{k} c_i \times t_i$ where c_i are integers and t_i are distinct non-decomposable terms without integer coefficients.

An ODE system is in additive normal form if each equation is in additive normal form, i.e. if it is of the form

$$\dot{x}_i = \sum_{j=1}^{l} c_{i,j} \times t_j, \quad 1 \leq i \leq s$$

where l is the number of non-decomposable terms t_j in the system.

Additive normal forms are not unique, but any ODE system can be written in additive normal form through standard algebraic transformations (such as the distributivity of $\times$ over $+$). The non-decomposability condition excludes the composition of several reactions in a single one with a sum as kinetic expression. In particular, we have:

Proposition 3.5. *Any non-decomposable well-founded reaction system, such that its ODE semantics is a polynomial ODE system, is strict.*

Proof. First notice that a polynomial kinetics once in additive normal form results in a sum of monomials as non-decomposable terms. Now, from the second condition of well-formedness in Definition 2.5, for each reaction (r, m, p, f) we have $r(x_j) > 0$ implies $\exists \vec{x}, \partial f / \partial x_j(\vec{x}) > 0$, but since f is a monomial, this implies that f has degree at least 1 in x_j, and therefore that $f(x_1, \ldots, x_s) = 0$ when $x_j = 0$, i.e., (r, m, p, f) is strict. □

Now, given an ODE system in additive normal form, the following algorithm can infer an equivalent reaction system by sorting the terms of the ODEs, and creating one reaction for each term (formalized in Proposition 3.9 below). This algorithms requires checking the sign of a partial derivative, and as described in Section 4.1, such checks can be arbitrarily difficult for arbitrary mathematical expressions, but can be over-approximated. We thus assume given a test program (exact

Please cite this article in press as: F. Fages et al., Inferring reaction systems from ordinary differential equations, Theor. Comput. Sci. (2014), http://dx.doi.org/10.1016/j.tcs.2014.07.032

Doctopic: Theory of natural computing ARTICLE IN PRESS TCS:9840

or not) for testing the sign of partial derivatives: `partial_has_pos_val`(f,x) that answers if yes or no the partial derivative of the function f with respect to variable x takes a non-negative value for some input in $\mathbb{R}^s$, and such that $\exists \vec{y}\, \partial f/\partial x(\vec{y}) > 0 \Rightarrow$ `partial_has_pos_val`(f,x). For computability reasons, the reverse implication is not required. These tests are used in steps 4(c) and 4(d).

Algorithm 3.6 *(Reaction inference).*

input: ODE system O over variables for molecular concentrations,
`partial_has_pos_val` test
1. rewrite O into additive normal form,
2. compute the set $\mathcal{T}$ of all terms appearing in O,
3. let $R := \emptyset$,
4. for each non-decomposable term t in $\mathcal{T}$,
   (a) let $r := \emptyset$, $p := \emptyset$, $m := \emptyset$,
   (b) for each variable x where t occurs with integer coefficient c in $\dot{x}$ in O,
      i. if $c < 0$ then $r(x) := -c$,
      ii. if $c > 0$ then $p(x) := c$,
   (c) for each variable x such that $r(x) = 0$ and `partial_has_pos_val`(t,x),
      i. $r(x) := 1$,
      ii. $p(x) := p(x) + 1$,
   (d) for each variable x such that `partial_has_pos_val`$(-t,x)$,
      i. $m(x) := 1$,
   (e) $R := R \cup \{r/m \xrightarrow{t} p\}$,

output: reaction system R.

Example 3.7. The model of three reactions of Example 2.10 has one invariant: $pMPF + MPF$ is indeed a constant c (the sum of initial values of $pMPF$ and MPF) since $\dot{pMPF} + \dot{MPF} = 0$. One variable, e.g. $pMPF$, can thus be eliminated and replaced by $c - MPF$. This yields the following ODE system, where all k_i are positive:

$$\dot{MPF} = k_1 \times (c - MPF) \times Cdc25 - k_2 \times MPF \times Wee1$$
$$\dot{Wee1} = k_3/(k_4 + Clock)$$
$$\dot{Cdc25} = 0$$
$$\dot{Clock} = 0$$

When applied to this system, using the test for partial derivatives described in Section 4.1, Algorithm 3.6 infers the following reactions:

$$Cdc25 \xrightarrow{c \times k_1 \times Cdc25} Cdc25 + MPF$$
$$MPF + Cdc25 \xrightarrow{k_1 \times Cdc25 \times MPF} Cdc25$$
$$MPF + Wee1 \xrightarrow{k_2 \times MPF \times Wee1} Wee1$$
$$\emptyset/Clock \xrightarrow{k_3/(k_4+Clock)} Wee1$$

However, by applying first the hidden molecule inference Algorithm 3.1, a hidden molecular species $MPFi$ is introduced for the expression $c - MPF$. This hidden molecule corresponds to the linear invariant $MPFi + MPF = c$. We have

$$\dot{MPF_i} = -k_1 \times MPF_i \times Cdc25 + k_2 \times MPF \times Wee1$$

and when applied to this ODE system after the preprocessing step, Algorithm 3.6 now computes the correct reactions:

$$MPFi + Cdc25 \xrightarrow{k_1 \times MPFi \times Cdc25} MPF + Cdc25$$
$$MPF + Wee1 \xrightarrow{k_2 \times MPF \times Wee1} MPFi + Wee1$$
$$\emptyset/Clock \xrightarrow{k_3/(k_4+Clock)} Wee1$$

By counting the loops, one can easily check

Proposition 3.8 *(Time complexity). On an ODE system O in additive normal form, Algorithm 3.6 computes a reaction system in time $O(n \times t)$, where n is the number of variables and t is the number of non-decomposable terms in O.*

Please cite this article in press as: F. Fages et al., Inferring reaction systems from ordinary differential equations, Theor. Comput. Sci. (2014), http://dx.doi.org/10.1016/j.tcs.2014.07.032

Doctopic: Theory of natural computing ARTICLE IN PRESS TCS:9840

By executing symbolically the algorithm, one can similarly check that the result is characterized in mathematical terms by

Proposition 3.9 *(Inferred reactions). Given an ODE system in additive normal form with appearing terms $\mathcal{T} = \{f_1, \ldots, f_t\}$: $\dot{x}_i = \sum_{u=1}^{t} c_{i,u} \times f_u$ for $1 \leq i \leq s$. The reaction system inferred by Algorithm 3.6 is the set of non-decomposable reactions*

$$\{r_u / m_u \xrightarrow{f_u} p_u\}_{1 \leq u \leq t}$$

where

$$r_u = \sum_{\{i | c_{i,u} < 0\}} (-c_{i,u}) \times x_i + \sum_{\{i | c_{i,u} \geq 0,\ \texttt{partial\_has\_pos\_val}(f_u, x_i)\}} x_i,$$

$$p_u = \sum_{\{i | c_{i,u} > 0\}} c_{i,u} \times x_i + \sum_{\{i | c_{i,u} \geq 0,\ \texttt{partial\_has\_pos\_val}(f_u, x_i)\}} x_i$$

and m_u is the set of variables x such that `partial_has_pos_val`$(-f_u, x)$.

Theorem 3.10 *(Soundness). The ODE semantics of the reaction system inferred by Algorithm 3.6 from an ODE system O is equal to O.*

Proof. Let us suppose without loss of generality that $O = \{\dot{x}_i = \sum_{u=1}^{t} c_{i,u} \times f_u \mid 1 \leq i \leq s\}$ is in additive normal form. The inferred reaction system is the set

$$\{r_u / m_u \xrightarrow{f_u} p_u\}_{1 \leq u \leq t}$$

where

$$r_u = \sum_{\{i | c_{i,u} < 0\}} (-c_{i,u}) \times x_i + \sum_{\{i | c_{i,u} \geq 0,\ \texttt{partial\_has\_pos\_val}(f_u, x_i)\}} x_i,$$

$$p_u = \sum_{\{i | c_{i,u} > 0\}} c_{i,u} \times x_i + \sum_{\{i | c_{i,u} \geq 0,\ \texttt{partial\_has\_pos\_val}(f_u, x_i)\}} x_i,$$

and m_u is the set of variables y such that `partial_has_pos_val`$(-f_u, x_i)$.

The ODE system associated to these reactions is thus

$$\left\{\dot{x}_i = \sum_{u=1}^{t} \big(p_u(x_i) - r_u(x_i)\big) \times f_u\right\}_{1 \leq i \leq s} = \left\{\dot{x}_i = \sum_{u=1}^{t} c_{i,u} \times f_u\right\}_{1 \leq i \leq s} = O.$$

Note that it does not depend on the test `partial_has_pos_val`. □

Algorithm 3.6 always computes a non-decomposable reaction system with an equivalent associated ODE system but this reaction system may not be well-formed. In particular, step 3(b) adds a variable x to the reactants of the reactions even if x does not appear in the kinetic expression f of the reaction. Therefore the algorithm may infer reactions with reactants that do not occur in the kinetic expression, as required for instance by Example 2.9.

We can measure the completeness of the method by showing that, at least, if we start from a well-formed reaction model, generate the ODE semantics, and from the ODE system solely, infer back a reaction model, the algorithm does infer a well-formed reaction model.

First, it is clear that the algorithm infers non-decomposable kinetics (Proposition 3.9) in which any variable appearing in the kinetics appears in the reaction as either reactant (step 4(b), or step 4(c) for catalysts), inhibitor (step 4(d)) or both:

Proposition 3.11. *The reactions inferred by Algorithm 3.6 contain no reaction with a molecular species x appearing in the kinetic expression f with $\partial f / \partial x \neq 0$, and not appearing as a reactant or inhibitor.*

This proposition remains true even if the sets of variables for which the partial derivatives are positive (4c) or negative (4d) are over-approximated. However, for completeness an exact test is necessary.

Theorem 3.12 *(Weak completeness). When applied to the ODE semantics of a non-decomposable well-formed reaction system such that $\forall 1 \leq i \leq n$, $1 \leq j \leq s$, $\partial f_i / \partial x_j > 0 \Leftrightarrow$* `partial_has_pos_val`$(f_i, x_j)$*, Algorithm 3.6 does infer a non-decomposable well-formed reaction system. Furthermore, if the ODE system is polynomial, the inferred model is strict.*

Please cite this article in press as: F. Fages et al., Inferring reaction systems from ordinary differential equations, Theor. Comput. Sci. (2014), http://dx.doi.org/10.1016/j.tcs.2014.07.032

Doctopic: Theory of natural computing ARTICLE IN PRESS TCS:9840

Proof. Let us consider the ODEs associated to a well-formed non-decomposable reaction system $R = \{r_i/m_i \xrightarrow{f_i} p_i\}_{i=1,\ldots,n}$. The ODE system is of the form $O = \{\dot{x}_j = \sum_{i=1}^{n}(p_i(x_j) - r_i(x_j)) \times f_i \mid 1 \le i \le m\}$ which is an additive normal form after evaluation of the integers $p_i(x_j) - r_i(x_j)$. By Proposition 3.9, the inferred reaction system is $\{r'_i/m'_i \xrightarrow{f_i} p'_i\}_{1\le i\le n}$ where f_i is non-decomposable by hypothesis,

$$r'_i = \sum_{\{j \mid p_i(x_j) < r_i(x_j)\}} \big(r_i(x_j) - p_i(x_j)\big) \times x_j + \sum_{\{j \mid p_i(x_j) \ge r_i(x_j), \partial f_i/\partial x_j > 0\}} x_j$$

$$p'_i = \sum_{\{j \mid p_i(x_j) > r_i(x_j)\}} \big(p_i(x_j) - r_i(x_j)\big) \times x_j + \sum_{\{j \mid p_i(x_j) \ge r_i(x_j), \partial f_i/\partial x_j > 0\}} x_j,$$

and $m'_i = m_i$.

Now for any variable x_j, we have $x_j \in r'_i$ if and only if $x_j \in r_i$ since either $p_i(x_j) < r_i(x_j)$ or $\partial f_i/\partial x_j > 0$. Similarly $x_j \in p'_i$ if and only if $x_j \in p_i$ since either $p_i(x_j) > r_i(x_j)$ or $p_i(x_j) = r_i(x_j)$ and $\partial f_i/\partial x_j > 0$, These equalities between the sets (not multisets) of reactants, products and inhibitors suffice to show the well-formedness of the inferred reactions.

Strictness in the polynomial case follows from Proposition 3.5. □

Since we do not restrict ourselves to Mass Action kinetics, our algorithm may well infer reactions with other kinetic expressions in cases where purely Mass Action reactions were possible. This is an important difference between our algorithm and the previous algorithms, which are restricted to Mass Action kinetics [30–32]. Furthermore, even if we restrict to polynomial ODEs and Mass Action kinetics for reaction, further conditions are necessary to grant the unicity of the solution [32].

Example 3.13. For instance, given the ODE system

$$\dot{x} = -2kx = -\dot{y},$$

our algorithm infers the reaction

$$2 \times x \xrightarrow{k\times x} 2 \times y,$$

whereas a Mass Action kinetic reaction model for this system is

$$x \xrightarrow{2\times k\times x} y.$$

Furthermore, another Mass Action reaction system exists for this ODE system:

$$x \xrightarrow{k\times x} x + 2 \times y$$
$$x \xrightarrow{k\times x} \emptyset$$

4. Evaluation results on biomodels.net

The ability to infer a reaction system from ODEs can be turned into some automatic curation algorithm, as was done in Theorem 3.12, by inferring the reactions from the ODE semantics of a starting reaction system. In this section, we evaluate this form of curation on repository of structured models.

4.1. Computability issues

Since, like in SBML, we allow arbitrary mathematical expression for kinetic expressions, checking the well-formed conditions may raise arbitrary difficult symbolic computation problems. These conditions can be checked however by doing some approximations.

In our implementation in Biocham [1,2],[1] the `partial_has_pos_val` proceeds as follows: the kinetic expressions are first normalized as if they were polynomials, stopping when a non-polynomial operator (anything else than $+$, $-$ and $\times$) is found. For the polynomials, the exact computation of the sign of any partial derivative is easy. For the other terms, either they are recognized as a standard kinetics (like Hill functions) and once again the exact sign is extracted, or they are considered unknown and for any variable appearing we will assume that it is possible that $\partial f/\partial x$ becomes positive for some values, and negative for some values. This is a conservative over-approximation.

With these provisions, different syntactical conditions may indicate that a reaction is not well-formed. The conditions for a reaction to be ill-formed can be classified into three categories:

[1] http://lifeware.inria.fr/biocham/.

Please cite this article in press as: F. Fages et al., Inferring reaction systems from ordinary differential equations, Theor. Comput. Sci. (2014), http://dx.doi.org/10.1016/j.tcs.2014.07.032

Doctopic: Theory of natural computing ARTICLE IN PRESS TCS:9840

Table 1
Number of models having a "K not R", "R not K", or "negative kinetics" warning among the original 361 models of the curated part of biomodels.net, and among the reaction systems automatically inferred from their ODE semantics. "Any warning" reflects model for which there was at least one of the three warning.

	"K not R"	"R not K"	"Negative"	Any warning
Original	173	123	157	234 (64.81%)
Inferred	0	67	70	103 (28.53%)

1. "K not R" indicates that the concentration of a compound appears in the kinetic law of a reaction, but this compound is neither a reactant nor an inhibitor of the reaction;
2. "R not K" indicates that some compound is marked as reactant or inhibitor in a reaction, but does not appear in the kinetic expression;
3. "Negative" indicates that a kinetic expression may be negative with non-negative concentration values.

Indeed, in a well-formed reaction with kinetic expression f, if a species x is neither a reactant nor an inhibitor, then $\partial f/\partial x = 0$, hence x should not appear in the kinetic expression f. Similarly, if a species is a reactant or an inhibitor, then $\partial f/\partial x \neq 0$, so x should appear in f. Moreover, f should be non-negative.

These ill-formedness conditions are checked in Biocham using the previous approximations. They correspond to the warning messages that Biocham can raise when loading a reaction system.

4.2. Global analysis

The 424 models from the curated branch of the latest version (release 24) of the biomodels.net repository [35] were used as benchmark to test our reaction system inference algorithm, and compare the results with the original writing of the models in SBML. Out of those 424 models only 361 define *reactions* with proper *kineticLaws*. The other ones only describe systems through events and rules, or with no kinetic information, and thus have no ODE semantics.

Our curation algorithm reads the SBML model, extracts the corresponding ODE system and infers from it a new reaction system.

Table 1 summarizes the result of the procedure, as detected by Biocham warnings. Over the 361 reaction systems of the original curated part of biomodels.net with ODE semantics, our algorithm reveals *hidden molecules* in 58 models, 173 models with "K not R" warning, 123 models with "R not K" warning and 157 models with "negative kinetics" warning. Our algorithm is able to automatically curate the writing of these models with reactions by reducing the number of non-well-formed models with a warning by more than the half, from 65% to 29%.

As predicted by Proposition 3.11, the Algorithm 3.6 completely removes the "K not R" warnings. For the two other warnings, since the algorithm focuses on *non-decomposable* kinetics, it results in curated models quite close to the original ones, but does not tackle thoroughly the case of reactions with rates independent of some reactant, for the reasons illustrated in Example 2.9 of for any other reason. Therefore, 103 over 361 models remain with a non-well-formedness warning.

4.3. Model inconsistencies studied in [19]

In [19], the authors also scan the biomodels.net repository and report finding 5 inconsistencies: models 44, 93, 94, 143 and 151. Their diagnostics is as follows, some reaction fluxes become negative during the simulations of those models because of missing reversibility indications in models 93, 94 and 143. In the two first cases they report that adding the reverse reactions makes the models consistent, whereas for 143 it is also necessary to change some kinetic law. For model 151 they report a "missing step", but since the opposite reaction is part of the model, once again this amounts to adding a reverse reaction to an existing one. Finally, for model 44 they describe that the issue is that some kinetic expression does not depend on one of the reactants of the reaction, making it possible for that reactant's concentration to become negative.

For models 93, 94 and 151, which indeed are flagged by the "Negative" warning, our algorithm correctly adds the missing reverse reactions, directly from the kinetic expressions. The models automatically curated this way do not raise any warning at the end.

For model 44, the automatic curation allows us to get rid of a "K not R" warning by transforming the reaction $v3$

$$A + Y \xrightarrow{cytosol \times Vm3 \times A^4 \times Y^2 \times Z^4 / ((Ka^4 + A^4) \times ((Ky^2 + Y^2) \times (Kz^4 + Z^4)))} A + Z$$

into

$$Z + A + Y \longrightarrow 2 \times Z + A$$

with the same kinetics.

However, as expected, the "R not K" warning identified by Kaleta et al. remains, the obtained model is still not well-formed. The same happens with model 143 where indeed an "R not K" warning remains after automatic curation, in accordance with the earlier results.

Please cite this article in press as: F. Fages et al., Inferring reaction systems from ordinary differential equations, Theor. Comput. Sci. (2014), http://dx.doi.org/10.1016/j.tcs.2014.07.032

5. Conclusion

We have described an algorithm for trying to infer a meaningful reaction system from a system of ordinary differential equations. This algorithm is based on a general consistency condition between the kinetic expression and the structure of a reaction in terms of its reactants, products and inhibitors.

We have shown some general properties enjoyed by the influence graph of the Jacobian sign matrix associated to such well-formed reaction systems. These theoretical results militate for distinguishing between catalysts and inhibitors in the modifiers of a reaction, and for using structural analysis methods before fixing parameter values and going to simulations.

We have also evaluated the capability of our reaction inference algorithm to automatically curate the writing of ODE models with reactions by applying it to the ODE models generated from the SBML models of the curated part of biomodels.net. In particular, we have shown that the inference of well-formed reactions from the ODEs, combined with the inference of hidden molecules corresponding to linear invariants, is sufficient to automatically curate the writing of some ODE models of the cell cycle with consistent reactions. On the whole curated part of the biomodels.net repository, we have shown that our automatic curation method significantly improves the writing of the models with reactions by reducing the number of non-well-formed reaction systems from 65% to 29%.

Although the primary concern of SBML is to provide a common format for exchanging models and doing simulations, we believe that stronger consistency conditions should be enforced in SBML to perform structural analyses, and that the strict well-formedness conditions presented in this paper should be verified by non-reversible reactions.

Acknowledgements

This work has been supported by the French OSEO Biointelligence and ANR Biotempo (ANR-10-BLANC-0218) projects. We are grateful to the partners of these projects for fruitful discussions. We also acknowledge the comments of the reviewers which were useful to improve the presentation of this article.

References

[1] F. Fages, S. Soliman, Formal cell biology in BIOCHAM, in: M. Bernardo, P. Degano, G. Zavattaro (Eds.), 8th Int. School on Formal Methods for the Design of Computer, Communication and Software Systems, Computational Systems Biology SFM'08, in: Lecture Notes in Comput. Sci., vol. 5016, Springer-Verlag, Bertinoro, Italy, 2008, pp. 54–80.
[2] L. Calzone, F. Fages, S. Soliman, BIOCHAM: an environment for modeling biological systems and formalizing experimental knowledge, Bioinformatics 22 (14) (2006) 1805–1807, http://dx.doi.org/10.1093/bioinformatics/btl172.
[3] R. Thomas, A.-M. Gathoye, L. Lambert, A complex control circuit: regulation of immunity in temperate bacteriophages, Eur. J. Biochem. 71 (1) (1976) 211–227.
[4] R. Thomas, On the relation between the logical structure of systems and their ability to generate multiple steady states or sustained oscillations, Springer Ser. Synergetics 9 (1981) 180–193.
[5] R. Thomas, Regulatory networks seen as asynchronous automata: a logical description, J. Theoret. Biol. 153 (1991) 1–23.
[6] S. Eker, M. Knapp, K. Laderoute, P. Lincoln, J. Meseguer, M.K. Sönmez, Pathway logic: symbolic analysis of biological signaling, in: Proceedings of the Seventh Pacific Symposium on Biocomputing, 2002, pp. 400–412.
[7] G. Bernot, J.-P. Comet, A. Richard, J. Guespin, A fruitful application of formal methods to biological regulatory networks: extending Thomas' asynchronous logical approach with temporal logic, J. Theoret. Biol. 229 (3) (2004) 339–347.
[8] N. Chabrier-Rivier, M. Chiaverini, V. Danos, F. Fages, V. Schächter, Modeling and querying biochemical interaction networks, Theoret. Comput. Sci. 325 (1) (2004) 25–44.
[9] F. Fages, S. Soliman, N. Chabrier-Rivier, Modelling and querying interaction networks in the biochemical abstract machine BIOCHAM, J. Biol. Phys. Chem. 4 (2) (2004) 64–73.
[10] L. Calzone, N. Chabrier-Rivier, F. Fages, S. Soliman, Machine learning biochemical networks from temporal logic properties, in: G. Plotkin (Ed.), Transactions on Computational Systems Biology VI, in: Lect. Notes in Bioinform., vol. 4220, Springer-Verlag, 2006, pp. 68–94, cMSB'05 Special Issue.
[11] V.N. Reddy, M.L. Mavrovouniotis, M.N. Liebman, Petri net representations in metabolic pathways, in: L. Hunter, D.B. Searls, J.W. Shavlik (Eds.), Proceedings of the 1st International Conference on Intelligent Systems for Molecular Biology, ISMB, AAAI Press, 1993, pp. 328–336.
[12] I. Zevedei-Oancea, S. Schuster, Topological analysis of metabolic networks based on Petri net theory, In Silico Biol. 3 (29) (2003).
[13] D. Angeli, P.D. Leenheer, E.D. Sontag, A Petri net approach to persistence analysis in chemical reaction networks, in: Biology and Control Theory: Current Challenges, in: Lecture Notes in Control and Inform. Sci., vol. 357, Springer-Verlag, 2007, pp. 181–216.
[14] C. Chaouiya, E. Remy, D. Thieffry, Petri net modelling of biological regulatory networks, J. Discrete Algorithms 6 (2) (2008) 165–177, http://dx.doi.org/10.1016/j.jda.2007.06.003.
[15] C. Rohr, W. Marwan, M. Heiner, Snoopy – a unifying Petri net framework to investigate biomolecular networks, Bioinformatics 26 (7) (2010) 974–975, http://dx.doi.org/10.1093/bioinformatics/btq050.
[16] S. Soliman, Invariants and other structural properties of biochemical models as a constraint satisfaction problem, Algorithms Mol. Biol. 7 (15) (2012), http://dx.doi.org/10.1186/1748-7188-7-15.
[17] S. Gay, S. Soliman, F. Fages, A graphical method for reducing and relating models in systems biology, Bioinformatics 26 (18) (2010) i575–i581, http://dx.doi.org/10.1093/bioinformatics/btq388, Special Issue ECCB'10.
[18] S. Gay, F. Fages, T. Martinez, S. Soliman, C. Solnon, On the subgraph epimorphism problem, Discrete Appl. Math. 162 (2014) 214–228, http://dx.doi.org/10.1016/j.dam.2013.08.008.
[19] C. Kaleta, S. Richter, P. Dittrich, Using chemical organization theory for model checking, Bioinformatics 25 (15) (2009) 1915–1922, http://dx.doi.org/10.1093/bioinformatics/btp332.
[20] S. Schuster, D.A. Fell, T. Dandekar, A general definition of metabolic pathways useful for systematic organization and analysis of complex metabolic networks, Nat. Biotechnol. 18 (2002) 326–332.
[21] A. Varma, B. Palsson, Metabolic flux balancing: basic concepts, scientific and practical use, Nat. Biotechnol. 12 (10) (1994) 994–998.
[22] P. Dittrich, P. di Fenizio, Chemical organisation theory, Bull. Math. Biol. 69 (4) (2007) 1199–1231, http://dx.doi.org/10.1007/s11538-006-9130-8.

Please cite this article in press as: F. Fages et al., Inferring reaction systems from ordinary differential equations, Theor. Comput. Sci. (2014), http://dx.doi.org/10.1016/j.tcs.2014.07.032

[23] M. Feinberg, Mathematical aspects of mass action kinetics, in: L. Lapidus, N.R. Amundson (Eds.), Chemical Reactor Theory: A Review, Prentice-Hall, 1977, pp. 1–78, Chap. 1.
[24] G. Shinar, M. Feinberg, Structural sources of robustness in biochemical reaction networks, Science 327 (5971) (2010) 1389–1391, http://dx.doi.org/10.1126/science.1183372.
[25] G. Koh, H. Teong, M.-V. Clement, D. Hsu, P. Thiagarajan, A decompositional approach to parameter estimation in pathway modeling: a case study of the akt and mapk pathways and their crosstalk, Bioinformatics 22 (14) (2006) e271–e280.
[26] S. Soliman, Finding minimal P/T-invariants as a CSP, in: Proceedings of the Fourth Workshop on Constraint Based Methods for Bioinformatics, WCB'08, 2008, Associated to CPAIOR'08.
[27] E. Grafahrend-Belau, F. Schreiber, M. Heiner, A. Sackmann, B.H. Junker, S. Grunwald, A. Speer, K. Winder, I. Koch, Modularization of biochemical networks based on a classification of petri net by T-invariants, BMC Bioinformatics 9 (90) (2008).
[28] F. Nabli, S. Soliman, Steady-state solution of biochemical systems, beyond S-systems via T-invariants, in: P. Quaglia (Ed.), CMSB'10: Proceedings of the 8th International Conference on Computational Methods in Systems Biology, CoSBi, ACM, 2010, pp. 14–22.
[29] D.T. Gillespie, Exact stochastic simulation of coupled chemical reactions, J. Phys. Chem. 81 (25) (1977) 2340–2361.
[30] V. Hárs, J. Tóth, On the inverse problem of reaction kinetics, in: M. Farkas (Ed.), Qualitative Theory of Differential Equations, in: Colloquia, vol. 30, 1979, pp. 363–379.
[31] G. Szederkényi, J.R. Banga, A.A. Alonso, Inference of complex biological networks: distinguishability issues and optimization-based solutions, BMC Syst. Biol. 5 (1) (2011) 177+, http://dx.doi.org/10.1186/1752-0509-5-177.
[32] S. Soliman, M. Heiner, A unique transformation from ordinary differential equations to reaction networks, PLoS One 5 (12) (2010) e14284, http://dx.doi.org/10.1371/journal.pone.0014284.
[33] M. Hucka, et al., The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models, Bioinformatics 19 (4) (2003) 524–531.
[34] M. Hucka, S. Hoops, S.M. Keating, L.N. Nicolas, S. Sahle, D. Wilkinson, Systems biology markup language (SBML) level 2: structures and facilities for model definitions, Nat. Precedings (2008), http://dx.doi.org/10.1038/npre.2008.2715.1.
[35] N. le Novère, B. Bornstein, A. Broicher, M. Courtot, M. Donizelli, H. Dharuri, L. Li, H. Sauro, M. Schilstra, B. Shapiro, J.L. Snoep, M. Hucka, BioModels database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems, Nucleic Acids Res. 1 (34) (2006) D689–D691.
[36] F. Fages, S. Soliman, From reaction models to influence graphs and back: a theorem, in: Proceedings of Formal Methods in Systems Biology, FMSB'08, in: Lecture Notes in Comput. Sci., vol. 5054, Springer-Verlag, 2008.
[37] F. Fages, S. Soliman, Abstract interpretation and types for systems biology, Theoret. Comput. Sci. 403 (1) (2008) 52–70, http://dx.doi.org/10.1016/j.tcs.2008.04.024.
[38] S. Soliman, A stronger necessary condition for the multistationarity of chemical reaction networks, Bull. Math. Biol. 75 (11) (2013) 2289–2303, http://dx.doi.org/10.1007/s11538-013-9893-7.
[39] C. Soulé, Mathematical approaches to differentiation and gene regulation, C. R., Biol. 329 (2006) 13–20.
[40] C. Soulé, Graphic requirements for multistationarity, Complexus 1 (2003) 123–133.
[41] E.H. Snoussi, Necessary conditions for multistationarity and stable periodicity, J. Biol. Systems 6 (1998) 3–9, http://dx.doi.org/10.1142/S0218339098000042.
[42] J.-L. Gouzé, Positive and negative circuits in dynamical systems, J. Biol. Systems 6 (1998) 11–15.
[43] M. Kaufman, C. Soulé, R. Thomas, A new necessary condition on interaction graphs for multistationarity, J. Theoret. Biol. 248 (2007) 675–685.
[44] K.W. Kohn, Molecular interaction map of the mammalian cell cycle control and DNA repair systems, Mol. Biol. Cell 10 (8) (1999) 2703–2734.
[45] M. Katsumata, Graphic representation of botts-morales equation for enzyme-substrate-modifier system, J. Theoret. Biol. 36 (2) (1972) 327–338.
[46] T.S. Gardner, M. Dolnik, J.J. Collins, A theory for controlling cell cycle dynamics using a reversibly binding inhibitor, Proc. Natl. Acad. Sci. USA 95 (24) (1998) 14190–14195, http://dx.doi.org/10.1073/pnas.95.24.14190.
[47] T. Matsuo, S. Yamaguchi, S. Mitsui, A. Emi, F. Shimoda, H. Okamura, Control mechanism of the circadian clock for timing of cell division in vivo, Science 302 (5643) (2003) 255–259.
[48] J.J. Tyson, Modeling the cell division cycle: cdc2 and cyclin interactions, Proc. Natl. Acad. Sci. USA 88 (16) (1991) 7328–7332.
[49] A. Goldbeter, A minimal cascade model for the mitotic oscillator involving cyclin and cdc2 kinase, Proc. Natl. Acad. Sci. USA 88 (20) (1991) 9107–9111.

Please cite this article in press as: F. Fages et al., Inferring reaction systems from ordinary differential equations, Theor. Comput. Sci. (2014), http://dx.doi.org/10.1016/j.tcs.2014.07.032

4.2.2 A Unique Transformation from ODEs to Reaction Networks

A more theoretical account on the same question is to look for mathematical criteria that ensure the unicity of the reaction model corresponding to an ODE system. This involves restricting the models looked for with even more stringent conditions (basically Mass-Action kinetics, with unique parameters) but allows the modeller to avoid any ambiguity in his translation.

[20] Sylvain Soliman and Monika Heiner. "A Unique Transformation from Ordinary Differential Equations to Reaction Networks". In: *PLoS One* 5.12 (Dec. 2010), e14284. DOI: 10.1371/journal.pone.0014284

OPEN ACCESS Freely available online

PLoS one

A Unique Transformation from Ordinary Differential Equations to Reaction Networks

Sylvain Soliman[1]*, **Monika Heiner**[2]

1 Equipe-Projet Contraintes, INRIA Paris-Rocquencourt, BP105, Paris, France, **2** Department of Computer Science, Brandenburg University of Technology, Cottbus, Germany

Abstract

Many models in Systems Biology are described as a system of Ordinary Differential Equations, which allows for transient, steady-state or bifurcation analysis when kinetic information is available. Complementary structure-related qualitative analysis techniques have become increasingly popular in recent years, like qualitative model checking or pathway analysis (elementary modes, invariants, flux balance analysis, graph-based analyses, chemical organization theory, etc.). They do not rely on kinetic information but require a well-defined structure as stochastic analysis techniques equally do. In this article, we look into the structure inference problem for a model described by a system of Ordinary Differential Equations and provide conditions for the uniqueness of its solution. We describe a method to extract a structured reaction network model, represented as a bipartite multigraph, for example, a continuous Petri net (CPN), from a system of Ordinary Differential Equations (ODEs). A CPN uniquely defines an ODE, and each ODE can be transformed into a CPN. However, it is not obvious under which conditions the transformation of an ODE into a CPN is unique, that is, when a given ODE defines exactly one CPN. We provide biochemically relevant sufficient conditions under which the derived structure is unique and counterexamples showing the necessity of each condition. Our method is implemented and available; we illustrate it on some signal transduction models from the BioModels database. A prototype implementation of the method is made available to modellers at http://contraintes.inria.fr/~soliman/ode2pn.html, and the data mentioned in the "Results" section at http://contraintes.inria.fr/~soliman/ode2pn_data/. Our results yield a new recommendation for the import/export feature of tools supporting the SBML exchange format.

Citation: Soliman S, Heiner M (2010) A Unique Transformation from Ordinary Differential Equations to Reaction Networks. PLoS ONE 5(12): e14284. doi:10.1371/journal.pone.0014284

Editor: Jean Peccoud, Virginia Tech, United States of America

Received May 7, 2010; **Accepted** November 15, 2010; **Published** December 22, 2010

Funding: SS received support from French ANR SYSCOMM 2008 project Calamar (ANR-08-SYSC-003). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

* E-mail: Sylvain.Soliman@inria.fr

Introduction

Many models in Systems Biology are described as a system of Ordinary Differential Equations (ODEs), which allows for transient and steady-state analysis (for instance using MATLAB®), or bifurcation analysis with tools like XPPAUT [1], but only when kinetic information is available.

Complementary structure-related qualitative analysis techniques have become increasingly popular in recent years, such as qualitative model checking or pathway analysis. Qualitative analysis techniques do not rely on kinetic information, but require a precisely structured model with well-identified products, reactants and catalysts (and their stoichiometry, if any) for each reaction.

The fact that the Systems Biology Markup Language (SBML) [2] has become a standard for sharing and publishing of models has helped in making modelers clarify the structure of their models. Unfortunately, SBML does not enforce that the structure and underlying ODEs are coherent. Even if the system is specified by a list of reactions, as supported, e.g., by COPASI [3], modelers tend to specify their reaction kinetics differently when aiming at ODEs analysis. The troublemakers are reactions with complex kinetics. COPASI provides a list of predefined functions; some of them actually stand for whole building blocks. Thus, the structural interpretation of models specified in formalisms such as SBML may vary according to the source of the original model. Particularly, if the models were originally meant to be ODE-oriented, a later discrete interpretation as a qualitative or stochastic model by a naive automatic translation may produce wrong results; see Figure 1 for an introductory example demonstrating the problem.

In [4], it is elaborated that structural information hidden in kinetic laws may affect the results obtained from structural analysis, such as elementary mode analysis [5], extreme pathway analysis [6], flux balance analysis [7], chemical organization theory [8], deficiency analysis or chemical reaction network theory (CRNT) [9,10]. This perfectly coincides with our own experience, and applies equally for place and transition invariant analysis to validate a model, see e.g. [11–13], or to derive automatically an hierarchically structured network representation [14].

Structural analysis may directly support ODEs-oriented dynamic analyses; e.g. [15] applies network decomposition for a modular parameter estimation approach, [16] introduces a structural persistency criterion, and transition invariants are used in [17] to identify fragile nodes and the core network responsible for the steady state behaviour, and in [18] to determine steady state solutions.

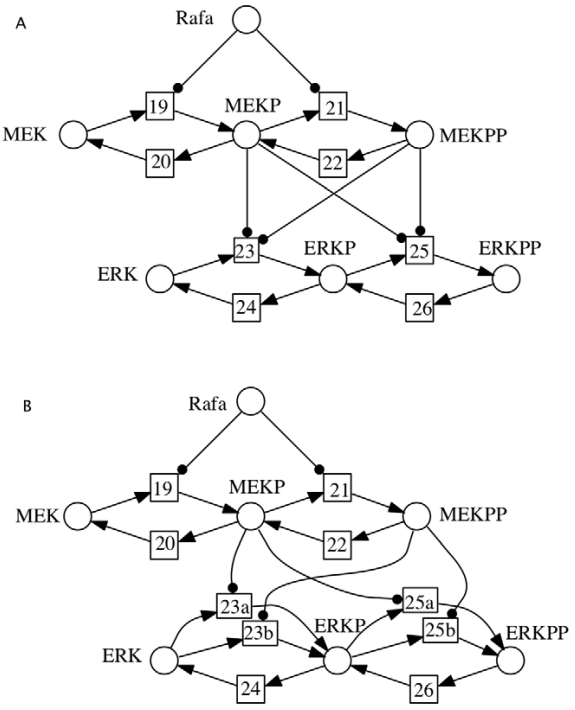


Figure 1. Arbitrary complex kinetics may hide essential structure. The example is an excerpt from the network model discussed in [33]. (A) Structure as suggested by the schematic representation in [33] and the list of reactions in the model's SBML format (Created by COPASI version 4.0 (Build 18) on 2006-10-24); (B) Correct structure, which is hidden in the kinetics of reactions 23 and 25. The two structures obviously differ in their discrete behaviour.
doi:10.1371/journal.pone.0014284.g001

Likewise, the correct structure is mandatory when a reaction network is meant to be put into a stochastic setting, as it has been introduced in the Petri net context in the seminal paper [19], and exercised by applying various stochastic analysis techniques (standard Markovian transient and steady state analysis, analytical and simulative model checking) to a running case study in, e.g., [12,20].

In [4], the authors present an algorithm that uncovers hidden structural information for some models already given in SBML. On the contrary, in our article we discuss conditions for unique structure inference directly from a given system of ODEs. We derive from those conditions an algorithm, that has been implemented and made public. We illustrate the necessity of our conditions and the result of the inference on some simple examples. This allows for a correct and automatic translation from ODE models to structured models suitable for qualitative or stochastic analysis, which we demonstrate on the very examples of the BioModels database [21] that were incorrectly transcribed in SBML as shown by [4].

We model a reaction network by a continuous Petri net (CPN), see [22]. We define $\mathcal{P}$, the set of places, with $n=|\mathcal{P}|$, and $\mathcal{T}$, the set of transitions, with $m=|\mathcal{T}|$. F^- and F^+ are $n\times m$ incidence matrices describing the weights of the transitions' input and output arcs, respectively. The matrix entries are denoted by f_{ij}^+ and f_{ij}^-, respectively.

Each transition $t\in\mathcal{T}$ has a rate function v_t specifying the generally state-dependent continuous flow over its input and output arcs. v_t can be an arbitrary function, but its variables are restricted to the pre-places of t to enforce a close relation between structure and dynamic behavior. A CPN uniquely defines a system of ODEs over the variables corresponding to the places $p_i\in\mathcal{P}$:

$$\frac{dp_i}{dt}=\sum_{j=1}^{m}(f_{ij}^+-f_{ij}^-)\cdot v_j \qquad (1)$$

We are interested in mapping a system of ODEs onto a CPN, such that the reverse operation according to (1) gives an equivalent system (up to simple algebraic operations obviously ensuring behavioral equivalence, such as $a\times v-b\times v=(a-b)\times v$). Thus, we will assume that the variables of the system of ODEs are: $x_i, 1\le i\le n$, i.e., each variable is mapped in a unique way to a place p_i of the net, which is required by the reverse mapping.

Such mappings have already been used in the Systems Biology community, e.g. in the need for a stochastic view of models originally described by ODEs. For instance in STODE [23], which was supposed to be included in COPASI, in BlenX [24], and the Beta Workbench [25]. However, no precise algorithm is described, and program sources of implementations are not available. Most importantly, these computational platforms do not care about our main concern – the uniqueness of the revealed structure.

Please note that any ODEs can be represented by a CPN simply by considering the full expression of each dx/dt, i.e. the right-hand side of the equation, as the v_x of a single transition t_x with all variables used in v_x (i.e., the domain of v_x) as pre-places, and exactly the same post-places (with the same arc weights), except for x itself, which should have as weight on $t_x\rightarrow x$ one more than the weight on $x\rightarrow t_x$; compare Figure 2. This naive translation always works and produces a net having an equal number of places and transitions, with structural information typically hidden in the generally complex kinetics v_x. However, it is not obvious under which conditions there is exactly one CPN corresponding to a system of ODEs (even if we assume minimal arc weights), and especially whether certain biologically reasonable conditions on the CPN enforce its uniqueness. In the following we discuss ODEs conditions ensuring that there exists only one CPN; but it will almost never be the one we get by the naive translation.

Methods

We will first present a restricted form of our results and then discuss its generalization to other types of kinetics. We will give examples where even quite simple kinetics leads to ambiguity, i.e., several nets can generate the same system of ODEs.

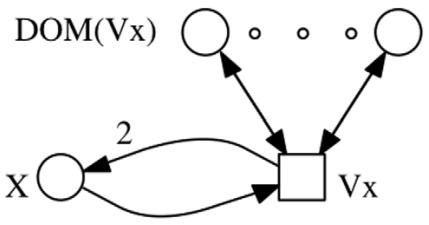


Figure 2. General principle to construct a CPN for an arbitrary ODEs. $DOM(v_x)$ denotes the domain of the function v_x.
doi:10.1371/journal.pone.0014284.g002

Mass Action Law

In order to obtain uniqueness of the net, we will first restrict ourselves to the case where our first condition holds.

Condition 1. *The CPN has pure mass action law kinetics, i.e.*

$$\forall j, 1 \leq j \leq m, v_j = k_j \cdot \prod_{i=1}^{n} x_i^{f_{ij}^-}$$

where the parameters k_j belong to a finite alphabet $\mathcal{K}$ of symbols.

Mass action is the basis of more elaborate rates used in biological models, like Michaelis-Menten or Hill kinetics, and the use of symbolic parameters is quite standard in ODEs models since it allows the modeler to "play" with a system of ODEs in a simple and coherent way. Mass action kinetics are also necessary for some stochastic simulation methods or analysis techniques like CRNT [9].

It is obvious that for arbitrary kinetics there is little hope to find a unique CPN. Moreover the following examples show that even quite simple kinetics can lead to ambiguity, i.e., several net structures can give the same system of ODEs (see Example 1), and that there is a need for symbolic parameters to ensure uniqueness (see Example 2).

Example 1. *Consider the following ODEs:*

$$\frac{dA}{dt} = -k \cdot A = -\frac{dB}{dt} \tag{2}$$

If one allows general kinetic expressions, even restricted such that they have the same variables as they have pre-places, one could obtain the two nets given in Figure 3.

Note that the second net does not respect Condition 1, since the kinetics should have been $k \cdot A^2$.

Example 2. *Consider the following ODEs:*

$$\frac{dA}{dt} = -2k \cdot A^2 = -\frac{dB}{dt} \tag{3}$$

Symbolic parameters are required to avoid that (3) *leads to the two nets given in Figure 4.*

We obtain the following system of ODEs by combining Condition 1 with equation (1):

$$\frac{dx_i}{dt} = \sum_{j=1}^{m} (f_{ij}^+ - f_{ij}^-) \cdot v_j = \sum_{j=1}^{m} (f_{ij}^+ - f_{ij}^-) \cdot k_j \cdot \prod_{h=1}^{n} x_h^{f_{hj}^-}, \forall i, 1 \leq i \leq n$$

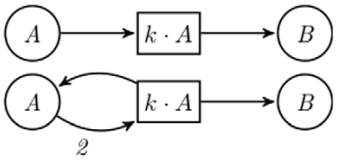


Figure 3. Two possible structures for Example 1. This illustrates the fact that arbitrary kinetic expressions introduce an ambiguity in the structure inference, even for very simple ODEs. The upper CPN represents the unique solution if reading equation (2) with the three established conditions.
doi:10.1371/journal.pone.0014284.g003

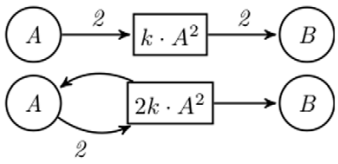


Figure 4. Two possible structures for Example 2. This illustrates the need for symbolic parameters in order to avoid confusion when inferring the structure.
doi:10.1371/journal.pone.0014284.g004

If a system of ODEs can be put in such a form, thanks to basic algebraic transformations, we will try to extract from it a CPN. Otherwise, it does obviously not correspond to any model fulfilling Condition 1.

We thus restrict our study to ODE systems of the form:

$$\frac{dx_i}{dt} = \sum_{j \in J} s_j \cdot l_j \cdot \prod_{h=1}^{n} x_h^{r_{ih}} \tag{4}$$

where J is a set of indices and for all $j \in J$ it holds $s_j \in \mathbb{Z}, l_j \in \mathcal{K}$, and all $r_{ih} \in \mathbb{N}$; in other words, ODE systems where the right side is a polynomial over x_i, with coefficients being integer linear combinations of parameters in $\mathcal{K}$.

A reaction which has exactly the same multisets of pre- and post-places, i.e., reactants and products, will only lead to null members in any ODE. Thus, we also assume:

Condition 2. *The CPN does not contain any void reaction, i.e.,*

$$\forall j, 1 \leq j \leq m, \exists i, 1 \leq i \leq n, f_{ij}^- \neq f_{ij}^+$$

Finally, we introduce a third purely syntactic condition to ensure uniqueness of the CPN.

Condition 3. *In the CPN, the same parameter is never used for two different reactions with the same reactants, i.e.,*

$$\forall j_1 j_2, 1 \leq j_1, j_2 \leq m, \begin{cases} either & k_{j_1} \neq k_{j_2} \\ or & \exists i, 1 \leq i \leq n, f_{ij_1}^- \neq f_{ij_2}^- \end{cases}$$

We illustrate Condition 3 by Example 3.

Example 3. *We consider again system* (2)*. Complying with Condition 1, but allowing a single parameter to be used twice for the same reactants, i.e., violating Condition 3, one could obtain the net given in Figure 5.*

Indeed, for the given system (2) and with the three introduced conditions, there are necessarily two places (A and B), one single transition (it has kinetics $k \cdot A$), a single pre-place (A with weight 1), and a single post-place (B with weight 1); see the first CPN of Example 1 in Figure 3.

Before turning to our main result, we introduce two lemmata.

Lemma 1. *Under our three conditions, all kinetics v_j appear at least once in the ODEs.*

Proof. Let us suppose that v_{j_0} does not appear in the system. We thus have $\exists J, \forall i, 1 \leq i \leq n, \sum_{j \in J} (f_{ij}^+ - f_{ij}^-) \cdot k_j \cdot \Pi_{h=1}^{n} x_h^{f_{hj}^-} = 0$ with $j_0 \in J$.

Let us first consider the case where $J = \{j_0\}$, i.e., the term $(f_{ij_0}^+ - f_{ij_0}^-) \cdot v_{j_0}$ amounts to 0 for all i. This would either violate Condition 1 if $v_{j_0} = 0$, or violate Condition 2 if $\forall i, f_{ij_0}^+ - f_{ij_0}^- = 0$.

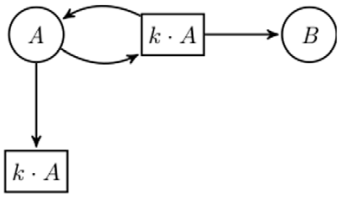


Figure 5. Another possible structure for the same equations as for Figure 3, as explained in Example 3. Even with symbolic parameters and pure mass action kinetics, if it is allowed to use the same parameter for two distinct reactions with the same reactants, one can obtain several structures for the same ODEs.
doi:10.1371/journal.pone.0014284.g005

Thus there are necessarily some terms compensating for v_{j_0} in some equations. These ODEs are precisely all the dx_i/dt such that $f^+_{ij_0} - f^-_{ij_0} \neq 0$.

However, since parameters are symbolic, only monomials with the same value of k_j and the same degree for all x_h can compensate each other. But under Condition 3 there are no other j that share these features with j_0.

Lemma 2. *Conversely, for each term $s \cdot l \cdot \prod x_h^{r_h}$ of the ODEs, there exists a transition with parameter l, and pre-places x_h with the corresponding arc weights r_h.*

Proof. The existence is obtained directly from the mapping of CPNs to ODEs according to (1). Since parameters and variables are symbolic objects, no term of that form can be created otherwise.

There is only a single such transition in any net agreeing with Condition 3. Thus, if there are several terms with the same l and r_h: $s_1 \cdot l \cdot \prod x_h^{r_h}, \ldots, s_q \cdot l \cdot \prod x_h^{r_h}$, they correspond to the same transition and can be merged into one single term $s \cdot l \cdot \prod x_h^{r_h}$ with $s = \sum_1^q s_i$.

We can now proceed to our main result.

Theorem 1. *For any system S of ODEs defining $dx_i/dt, 1 \leq i \leq n$* according to Conditions 1–3, there exists at most one CPN, such that the system S' obtained from it according to (1) *is equivalent to S, up to basic arithmetic.*

Proof. We have seen that the x_i uniquely define $\mathcal{P}$. From Lemma 1 and 2 we obtain the uniqueness of the definition of $\mathcal{T}$ and F^-.

Now, the post-places and corresponding weights are defined unambiguously by looking at dx_i/dt and imposing the constraint $s = f^+_{ij} - f^-_{ij}$, i.e., $f^+_{ij} = s + f^-_{ij}$ with f^-_{ij} already determined to be equal to some r_h in the previous step. If the obtained f^+_{ij} is strictly negative, there is no CPN that would produce such system under the assumed conditions.

The theorem states that there is *at most* one CPN. Indeed lots of ODEs are not amenable to (4) and thus do not comply with our first condition. However even for some systems that do comply with it there exists no model fulfilling our three conditions, as illustrated by Example 4.

Example 4. *An ODE system that can be put in the form of equation* (4), *but does not correspond to any CPN fulfilling our three conditions is $dx/dt = -2kx$.*

In this case, from the ODEs one would obtain a single place for x, a single transition with parameter k, an input weight of 1, but no possible output weight: $f^+ = s + f^- = -2 + 1 = -1$.

Beyond Mass Action Law

About 10% of the models of the BioModels database fulfill our three conditions. However it is quite common to use classical enzymatic kinetics like Michaelis-Menten or Hill type kinetics.

Actually, one can weaken Condition 1 in order to cope with Michaelian kinetics of the form: $v_j = \frac{V_j \cdot x_j}{K_j + x_j}$ in addition to the mass action law case.

Instead of polynomials, the right members of the ODEs will then be rational fractions. But thanks to the partial fraction decomposition theorem (see for instance [26]) they can be decomposed in a unique way into a sum of a polynomial and of rational fractions, with irreducible polynomials as denominator and a numerator of strictly smaller degree.

In our case, the simple rational fractions will have degree one denominator $(K_j + x_j)$ and degree zero numerator, otherwise there is no CPN corresponding to these ODEs without violating our new condition. These fractions can be easily and unequivocally transformed into the above form, the remaining polynomial will be handled as in the previous section.

Results

We built a prototype implementation of the method outlined above – the tool ode2pn, which converts XPPAUT files into SBML (Level 2, Version 1) or APNN (one of the standard Petri net formats [27]), respectively, by applying directly the constructive proof of Theorem 1. We built upon an already existing tool, Nicotine [28], for the output of the structured model and added to it an XPPAUT parser that uses Lemma 2 to introduce a new reaction for each corresponding term in the ODEs and Theorem 1 to complete the stoichiometry matrix.

The tool rejects the conversion when no structured model fulfilling our conditions can be obtained. It is available at http://contraintes.inria.fr/~soliman/ode2pn.html.

Note that the partial fraction decomposition necessary for the Michaelian kinetics always exists, but is "practical" only with prior knowledge of the poles of the denominator's polynomials. These are the K_j in the Michaelian case. Actually, our implementation supposes that the corresponding rational fractions are already in decomposed form.

In [4], five models from the BioModels database were identified as having been transcribed in SBML with some structural information missing: models 44, 93, 94, 143 and 151 (we adopt the convention to reduce the official model names to at most three digits). Model 44 involves Hill Kinetics and model 143 even more complex kinetic laws; so our approach cannot guarantee the uniqueness of the structure for these two cases. In the following we discuss our results for the remaining three models.

Contrary to [4], where SBML files are evaluated directly, we take the auto-generated XPP files (i.e. ODEs, generated from those SBML models), which we downloaded from the BioModels database in September 2009, and hand-curated in order to obtain exactly the ODEs as given in the original articles.

Models 93 and 94 are two models of the JAK/STAT pathway by [29]. In the original article they are described by a drawing (see Fig. 6) and a mixture of what the authors call "chemical reactions" and of ODEs (mostly for mRNAs). They are used as ODEs for simulation and were hand-transcribed to SBML for inclusion in BioModels database, but missing the "reversibility" of some reactions. We input the 34 differential equations (in each case) to our tool, with sometimes more than ten different terms in a single equation, and obtained the unique structure complying with our conditions (with the Michaelian extension) and correctly including reverse reactions when needed.

Model 151 is a model of the regulation of that same JAK/STAT pathway by IL-6 in hepatocytes [30]. It includes 68 differential equations (see Fig. 7 for an extract) and once again leads to a unique structure (with mass action and Michaelian

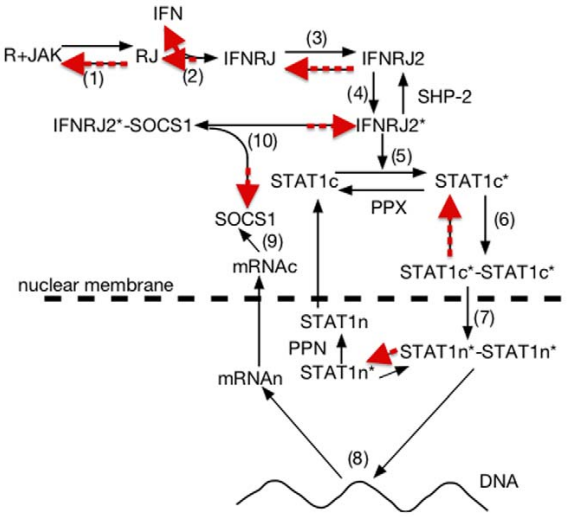


Figure 6. Figure 1 of [29] representing a schematic view of the JAK/STAT pathway. The incorrect structure of the corresponding SBML models (93 and 94) of the BioModels database can be automatically fixed by going back to the differential equations and extracting the unique structure fulfilling our three conditions. It then correctly includes the reversibility of reactions (1), (2), (3), (6), etc. highlighted in red, and absent from the BioModels database version.
doi:10.1371/journal.pone.0014284.g006

kinetics). The XPPAUT.ode file (BIOMD151.ode) and the resulting structured SBML file (BIOMD151_new.xml) can be found at http://contraintes.inria.fr/~soliman/ode2pn_data/ together with the biomodels version (BIOMD00000000151.xml), which actually contains more errors than found by [4], mostly concerning parameter names that are quite error-prone when hand-translated from ODEs to SBML. Note that the XPPAUT file which we provide corrects two typos from the original article, namely k_{r39} instead of k_{r30} in dx_8/dt and x_{15} instead of x_{14} in dx_{16}/dt. These typos still allow extraction of a unique structured model, but with obvious differences compared to that described in the article.

The converted models can be further processed by any tool complying with SBML or APNN, e.g. using Snoopy [31], which supports both formats and allows for graphical visualization of the translation results.

Discussion

We have discussed conditions for a unique structure inference out of a given system of ODEs. For reaction networks fulfilling the given three conditions, ODEs and a structured formalization by, e.g., a CPN, are equivalent representations, which can be transformed into each other without loss of information. Note that these networks are restricted to mass action or Michaelian kinetics, which are the most widely used kinetics for biochemical systems, and prohibit empty reactions which would not have any biochemical meaning. These conditions forbid models, which were mathematically correct, but contradict reasonable biochemical expectations.

We have shown that otherwise the structure is not uniquely defined by a system of ODEs. We have given examples where violating our conditions leads to several nets having possibly different discrete, and thus stochastic behavior, but generating the same system of ODEs. These counterexamples demonstrate the

$$\frac{dx_6}{dt} = k_{f2}x_5x_2 + 2k_{r3}x_7 - k_{r2}x_6 - 2k_{f3}x_6^2$$
$$\frac{dx_7}{dt} = k_{10}x_{16} + k_{10}x_{32} + k_{f3}x_6^2 - k_{r3}x_7 - k_4x_7$$
$$\frac{dx_8}{dt} = k_4x_7 + k_6x_{11} + k_{f32}x_{41} + k_{f37}x_{39} + k_{f39}x_{40} + k_{r21}x_{30} + k_{r5}x_{11} + k_{r7}x_{12} + k_{r9}x_{16} - k_{r39}x_8x_{45} - k_{r37}x_{46}x_8 - k_{r32}x_8x_{44} - k_{f9}x_{15}x_8 - k_{f7}x_{10}x_8 - k_{f5}x_9x_8 - k_{f21}x_8x_{29}$$
$$\frac{dx_9}{dt} = k_{10}x_{32} + k_{12}x_{18} + k_{17}x_{22} + k_{r13}x_{14} + k_{r5}x_{11} + k_{r5}x_{31} - k_{f5}x_9x_8 - k_{f5}x_{30}x_9 - k_{f13}x_{10}x_9$$
$$\frac{dx_{10}}{dt} = k_6x_{11} + k_{r11}x_{18} + k_{r13}x_{14} + k_{r7}x_{12} + 2k_{r8}x_{13} - 2k_{f8}x_{10}^2 - k_{f7}x_{10}x_8 - k_{f13}x_{10}x_9 - k_{f11}x_{17}x_{10}$$
$$\frac{dx_{11}}{dt} = k_{f5}x_9x_8 - k_{r5}x_{11} - k_6x_{11}$$
$$\frac{dx_{12}}{dt} = k_{f7}x_{10}x_8 - k_{r7}x_{12}$$
$$\frac{dx_{13}}{dt} = k_{f8}x_{10}^2 + k_{r11}x_{19} - k_{r8}x_{13} - k_{f11}x_{17}x_{13} - k_{14}x_{13}$$
$$\frac{dx_{14}}{dt} = k_{12}x_{19} + k_{f13}x_{10}x_9 - k_{r13}x_{14}$$
$$\frac{dx_{15}}{dt} = k_{10}x_{16} + k_{10}x_{32} + k_{r9}x_{16} + k_{r9}x_{32} + \frac{Vm}{Km + x_{46}}x_{46} - k_{f9}x_{31}x_{15} - k_{f9}x_{15}x_8$$

Figure 7. Beginning of the Appendix II of [30] describing the full ODE model of that article. The 68 ODEs actually allow the extraction of a unique model fulfilling the three established conditions. It not only correctly reflects the structure described in the article, but also avoids the typos introduced in the hand-written model 151 of the BioModels database; hand-typing an SBML model for that many ODEs with numerous parameters is definitely error-prone.
doi:10.1371/journal.pone.0014284.g007

necessity of each individual condition. We have given a constructive proof for the translation algorithm, which has been directly implemented, providing XPP to SBML conversion.

Our conditions are quite restrictive (only Mass-Action and Michaelian kinetics), but do cover a large part of mathematical biology models. This should allow, in the future, more and more modelers to benefit from structural analysis techniques for their systems, even if done as an afterthought. It also leads to more precise links between the different formalisms and launches a bridge betweens different communities of the Systems Biology field. In those cases where both the ODEs and a reaction diagram are given, our method allows the check if they are consistent.

Ideally, models are specified with our conditions in mind, be it as a list of reactions (as, e.g., in COPASI) or some graphical notation (e.g., continuous Petri nets). In both cases, kinetic functions should obey the three established conditions. User-friendly tools might check these conditions while doing export to SBML files to prevent misleading results by later use. Sophisticated ODE tools will have no problems in applying adequate algebraic transformations to optimize the simulation algorithms' run-time behavior. Any import of SBML files should check these conditions if aiming at structure-related qualitative or stochastic analysis techniques.

We intend to continue in trying to find uniqueness conditions for more general kinetics, and to devise heuristics for structure inference when uniqueness cannot be obtained (unwinding algebraic conservation laws coming from rapid equilibria, for instance). We also plan to make our algorithm more widely usable, for instance through a CellDesigner [32] XPP-import plugin.

Author Contributions

Conceived and designed the experiments: MEH. Performed the experiments: SS. Analyzed the data: SS. Wrote the paper: SS MEH.

References

1. Ermentrout B (2002) Simulating, Analyzing, and Animating Dynamical Systems: A Guide to XPPAUT for Researchers and Students. Philadelphia: SIAM, Available: http://www.math.pitt.edu/bard/xpp/xpp.html.
2. Hucka M, Finney A, Sauro HM, Bolouri H, Doyle JC, et al. (2003) The systems biology markup language (SBML): A medium for representation and exchange of biochemical network models. Bioinformatics 19: 524–531.
3. Hoops S, Sahle S, Gauges R, Lee C, Pahle J, et al. (2006) Copasi – a complex pathway simulator. BioInformatics 22: 3067–3074.
4. Kaleta C, Richter S, Dittrich P (2009) Using chemical organization theory for model checking. Bioinformatics 25: 1915–1922.
5. Schuster S, Fell DA, Dandekar T (2002) A general definition of metabolic pathways useful for systematic organization and analysis of complex metabolic networks. Nature Biotechnology 18: 326–332.
6. Palsson B (2006) Systems Biology: Properties of Reconstructed Networks. Cambridge: Cambridge University Press.
7. Varma A, Palsson B (1994) Metabolic flux balancing: basic concepts, scientific and practical use. Nature Biotechnology 12: 994–998.
8. Dittrich P, di Fenizio P (2007) Chemical organisation theory. Bulletin of Mathematical Biology 69: 1199–1231.
9. Feinberg M (1977) Mathematical aspects of mass action kinetics. In: Lapidus L, Amundson NR, eds. Chemical Reactor Theory: A Review Prentice-Hall. pp 1–78.
10. Shinar G, Feinberg M (2010) Structural sources of robustness in biochemical reaction networks. Science 327: 1389–1391.
11. Heiner M, Koch I (2004) Petri Net Based Model Validation in Systems Biology. In: Proc. 25th ICATPN 2004, LNCS 3099. Springer. pp 216–237.
12. Heiner M, Gilbert D, Donaldson R (2008) Petri nets in systems and synthetic biology. In: Schools on Formal Methods (SFM); LNCS 5016. Springer. pp 215–264.
13. Clark A, Gilmore S, Guerriero ML, Kemper P (2010) On verifying bio-pepa models. In: CMSB '10: Proceedings of the 8th International Conference on Computational Methods in Systems Biology. New York: ACM. pp 23–32. doi:http://doi.acm.org/10.1145/1839764.1839769.
14. Heiner M (2009) Understanding network behaviour by structured representations of transition invariants. In: Condon A, ed. Algorithmic Bioprocesses; Natural Computing series Springer. pp 367–389.
15. Koh G, Teong H, Clement MV, Hsu D, Thiagarajan P (2006) A Decompositional Approach to Parameter Estimation in Pathway Modeling: a Case Study of the Akt and MAPK Pathways and their Crosstalk. Bioinformatics 22: e271–e280.
16. Angeli D, De Leenheer P, Sontag E (2007) A Petri net approach to persistence analysis in chemical reaction networks. In: Biology and Control Theory: Current Challenges; LNCI 357 Springer. pp 181–216.
17. Heiner M, Sriram K (2010) Structural analysis to determine the core of hypoxia response network. PLoS ONE 5: e8600.
18. Nabli F, Soliman S (2010) Steady-state solution of biochemical systems, beyond s-systems via t-invariants. In: CMSB '10: Proceedings of the 8th International Conference on Computational Methods in Systems Biology. New York: ACM. pp 14–22. doi: http://doi.acm.org/10.1145/1839764.1839768.
19. Goss PJE, Peccoud J (1998) Quantitative modeling of stochastic systems in molecular biology by using stochastic Petri nets. Proc Natl Acad Sci USA 95: 2340–2361.
20. Heiner M, Donaldson R, Gilbert D (2010) Petri Nets for Systems Biology, In: Iyengar MS, ed. Symbolic Systems Biology: Theory and Methods Jones and Bartlett Publishers, Inc.
21. le Novère N, Bornstein B, Broicher A, Courtot M, Donizelli M, et al. (2006) BioModels Database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems. Nucleic Acid Research 1: D689–D691.
22. Gilbert D, Heiner M (2006) From Petri nets to differential equations - an integrative approach for biochemical network analysis. In: Proc. ICATPN 2006; LNCS 4024. Springer. pp 181–200.
23. Gend CV, Kummer U (2001) STODE - automatic stochastic simulation of systems described by differential equations. In: Yi TM, Hucka M, Morohashi M, Kitano H, eds. Proceedings of the 2nd International Conference on Systems Biology. Madison, WI: Omnipress. pp 326–333.
24. Dematté L, Priami C, Romanel A (2008) The BlenX language: A tutorial. In: Bernardo M, Degano P, Zavattaro G, eds. 8th Int. School on Formal Methods for the Design of Computer, Communication and Software Systems: Computational Systems Biology SFM'08; volume 5016 of *Lecture Notes in Computer Science*. Bertinoro, Italy: Springer-Verlag. pp 313–365.
25. Dematte L, Priami C, Romanel A (2008) The beta workbench: a computational tool to study the dynamics of biological systems. Brief Bioinform 9: 437–449.
26. Euler L (1809) De resolutione fractionum compositarum in simpliciores. Mémoires de l'Académie des sciences de St-Pétersbourg 1: 3–25.
27. Bause F, Kemper P, Kritzinger P (1994) Abstract Petri Net Notation. Technical report, Univ. Dortmund, CS Dep.
28. Soliman S (2009) Modelling biochemical reaction networks with biocham extracting qualitative and quantitative information from the structure. In: Proceedings of the 6th Vienna Conference on Mathematical Modelling MATHMOD'09. ARGESIM, volume 35: 2304–2312.
29. Yamada S, Shiono S, Joo A, Yoshimura A (2003) Control mechanism of jak/stat signal transduction pathway. FEBS Letters 534: 190–196.
30. Singh A, Jayaraman A, Hahn J (2006) Modeling regulatory mechanisms in il-6 signal transduction in hepatocytes. Biotechnology and Bioengineering 95: 850–862.
31. Rohr C, Marwan W, Heiner M (2010) Snoopy - a unifying Petri net framework to investigate biomolecular networks. Bioinformatics 26: 974–975.
32. Funahashi A, Matsuoka Y, Jouraku A, Morohashi M, Kikuchi N, et al. (2008) Celldesigner 3.5: A versatile modeling tool for biochemical networks. Proceedings of the IEEE 96: 1254–1265.
33. Brightman F, Fell D (2000) Differential feedback regulation of the MAPK cascade underlies the quantitative differences in EGF and NGF signalling in PC12 cells. FEBS letters 482: 169–174.

I have opinions of my own —strong opinions— but I don't always agree with them.

George Bush

Conclusion

The main objective of this compilation of articles was to try to answer some of the questions raised in the introduction, most notably about the nature of a model, as the main object of study in Systems Biology. The message we tried to convey is that what really is at the core of such a model is its drawing, its *structure*, as a bipartite multigraph, i.e. a reaction system. From there, through various analyses, we have shown that it is possible to reason on the dynamics of the system, if necessary in a qualitative way, and therefore to rule out in-silico procedures like parameter search or even wet-lab experiments, when they are vowed to fail, or unnecessary. Since the main job of a computational biologist is, or at least should be, to invalidate models, the gain is clear.

The structure is also what allows modellers to reason on huge models —as we see more and more— where precise kinetic data is missing, to compare different models and build some kind of phylogeny of them, to compose and reuse models, to relate different semantics of a same system. These are big steps in the direction of the Product Life Management view that might allow big pharmaceutical companies to adopt Systems Biology tools and methods. One of the crucial points for us was the use of Constraint Programming as a generic tool to tackle the discrete problems raised by all these applications, both for fast prototyping and for quite efficient procedures.

The previous point about semantics actually demonstrates that the structure is not opposed to the traditional view of mathematical biology that a model is a system of ODEs, on the contrary, it allows to reasonably build a system of reactions compatible with ODEs, or in the other direction to reduce a model enough to make it usable as a system of ODEs. As noted by the SBML effort, the meaning of a model is more conveyed by a reaction system than by a MATLAB® file.

The need for formal methods as analysis tools for more and more detailed models will probably increase in the coming years, but this will only translate into practical tools for biologists and modellers if we stop opposing those to the old-school analytical tools. Instead we should focus on making both work together in the best possible way. We believe that this is where computational Systems Biology has a big margin for improvement: reconcile the practical small-scale ODE-based users and the developers of formal methods tackling big structural models.

Some direction that we have not studied yet is that of influence models, which are also structural in some sense. Though we have encountered them, they were mostly seen as intermediary tools as in our condition for multistationarity. They are part of the hierarchy of semantics, but are "on the side". Recent works on influence systems [27] might make them more central and we hope to reconcile that way our big family of Systems Biology models.

Other topics that we plan to investigate further are the connexion between influence graph related circuit reasoning (i.e., our condition for multistationarity) and the loops encountered in CRNT and leading to similar type of results, but for a restricted class of models (usually Mass Action kinetics are required).

We have also started to work in collaboration with researchers from the computer algebra field in order to try and integrate into biochemical modelling environments like BIOCHAM their state of the arts results, which are much stronger than what most people in Systems Biology expect (e.g. formal computation of precise steady states or Hopf bifurcations for models of several tens of variables).

There are many fields to explore, both on the formal side and more in connexion with applications and modellers. We feel that Systems Biology is currently too fragmented to be as efficient as it could be and believe that working at the edge between different techniques and methods, and notably between structure and dynamics, while keeping an eye on practical issues, as we have started to demonstrate here, is the key to the successful evolution of the field.

Selected Publications

[1] Laurence Calzone, Nathalie Chabrier-Rivier, François Fages, and Sylvain Soliman. "Machine learning biochemical networks from temporal logic properties". In: *Transactions on Computational Systems Biology VI*. Ed. by Gordon Plotkin. Vol. 4220. Lecture Notes in BioInformatics. CMSB'05 Special Issue. Springer-Verlag, Nov. 2006, pp. 68–94. DOI: 10.1007/11880646_4.

[2] Laurence Calzone, François Fages, and Sylvain Soliman. "BIOCHAM: An Environment for Modeling Biological Systems and Formalizing Experimental Knowledge". In: *Bioinformatics* 22.14 (2006), pp. 1805–1807. DOI: 10.1093/bioinformatics/btl172.

[3] Katherine Chiang, François Fages, Jie-Hong Jiang, and Sylvain Soliman. "On the Hybrid Composition and Simulation of Heterogeneous Biochemical Models". In: *CMSB'13: Proceedings of the eleventh international conference on Computational Methods in Systems Biology*. Vol. 8130. Lecture Notes in BioInformatics. Springer-Verlag, Sept. 2013, pp. 192–205. DOI: 10.1007/978-3-642-40708-6_15.

[4] François Fages, Steven Gay, and Sylvain Soliman. "Inferring Reaction Models from ODEs". In: *CMSB'12: Proceedings of the tenth international conference on Computational Methods in Systems Biology*. Vol. 7605. Lecture Notes in BioInformatics. Springer-Verlag, Sept. 2012, pp. 370–373. DOI: 10.1007/978-3-642-33636-2_23.

[5] François Fages, Steven Gay, and Sylvain Soliman. "Inferring Reaction Systems from Ordinary Differential Equations". In: *Theoretical Computer Science* 599 (Sept. 2015), pp. 64–78. ISSN: 0304-3975. DOI: 10.1016/j.tcs.2014.07.032.

[6] François Fages and Sylvain Soliman. "Abstract Interpretation and Types for Systems Biology". In: *Theoretical Computer Science* 403.1 (2008), pp. 52–70. DOI: 10.1016/j.tcs.2008.04.024.

[7] François Fages and Sylvain Soliman. "From reaction models to influence graphs and back: a theorem". In: *Proceedings of Formal Methods in Systems Biology FMSB'08*. Lecture Notes in Computer Science 5054. Springer-Verlag, Feb. 2008. DOI: 10.1007/978-3-540-68413-8_7.

[8] François Fages, Sylvain Soliman, and Nathalie Chabrier-Rivier. "Modelling and Querying Interaction Networks in the Biochemical Abstract Machine BIOCHAM". In: *Journal of Biological Physics and Chemistry* 4.2 (Oct. 2004), pp. 64–73.

[9] Steven Gay, François Fages, Thierry Martinez, Sylvain Soliman, and Christine Solnon. "On the subgraph Epimorphism Problem". In: *Discrete Applied Mathematics* 162 (Jan. 2014), pp. 214–228. DOI: 10.1016/j.dam.2013.08.008.

[10] Steven Gay, Sylvain Soliman, and François Fages. "A Graphical Method for Reducing and Relating Models in Systems Biology". In: *Bioinformatics* 26.18 (2010). special issue ECCB'10, pp. i575–i581. DOI: 10.1093/bioinformatics/btq388.

[11] Faten Nabli, François Fages, Thierry Martinez, and Sylvain Soliman. "A Boolean Model for Enumerating Minimal Siphons and Traps in Petri-nets". In: *Proceedings of CP'2012, 18th International Conference on Principles and Practice of Constraint Programming*. Vol. 7514. Lecture Notes in Computer Science. Springer-Verlag, Oct. 2012, pp. 798–814. DOI: 10.1007/978-3-642-33558-7_57.

[12] Faten Nabli, Thierry Martinez, François Fages, and Sylvain Soliman. "On Enumerating Minimal Siphons in Petri nets using CLP and SAT solvers: Theoretical and Practical Complexity". In: *Constraints* 21.2 (2016), pp. 251–276. ISSN: 1383-7133. DOI: 10.1007/s10601-015-9190-1.

[13] Faten Nabli and Sylvain Soliman. "Steady-state solution of biochemical systems, beyond S-Systems via T-invariants". In: *CMSB'10: Proceedings of the 8th International Conference on Computational Methods in Systems Biology*. Ed. by Paola Quaglia. CoSBi. Trento, Italy: ACM, Oct. 2010, pp. 14–22. ISBN: 978-1-4503-0068-1. DOI: 10.1145/1839764.1839768.

[14] Aurélien Rizk, Grégory Batt, François Fages, and Sylvain Soliman. "On a Continuous Degree of Satisfaction of Temporal Logic Formulae with Applications to Systems Biology". In: *CMSB'08: Proceedings of the fourth international conference on Computational Methods in Systems Biology*. Ed. by Monika Heiner and Adeline Uhrmacher. Vol. 5307. Lecture Notes in Computer Science. Springer-Verlag, Oct. 2008, pp. 251–268. DOI: 10.1007/978-3-540-88562-7_19.

[15] Sylvain Soliman. "Finding minimal P/T-invariants as a CSP". In: *Proceedings of the fourth Workshop on Constraint Based Methods for Bioinformatics WCB'08, associated to CPAIOR'08*. May 2008.

[16] Sylvain Soliman. "Invariants and Other Structural Properties of Biochemical Models as a Constraint Satisfaction Problem". In: *Algorithms for Molecular Biology* 7.15 (May 2012). DOI: 10.1186/1748-7188-7-15.

[17] Sylvain Soliman. "A stronger necessary condition for the multistationarity of chemical reaction networks". In: *Bulletin of Mathematical Biology* 75.11 (Nov. 2013), pp. 2289–2303. DOI: 10.1007/s11538-013-9893-7.

[18] Sylvain Soliman, François Fages, and Ovidiu Radulescu. “A Constraint Solving Approach to Tropical Equilibration and Model Reduction”. In: *Proceedings of the ninth Workshop on Constraint Based Methods for Bioinformatics WCB'13, colocated with CP 2013*. Sept. 2013, pp. 27–36.

[19] Sylvain Soliman, François Fages, and Ovidiu Radulescu. “A constraint solving approach to model reduction by tropical equilibration”. In: *Algorithms for Molecular Biology* 9.24 (Dec. 2014). ISSN: 1748-7188. DOI: 10.1186/s13015-014-0024-2.

[20] Sylvain Soliman and Monika Heiner. “A Unique Transformation from Ordinary Differential Equations to Reaction Networks”. In: *PLoS One* 5.12 (Dec. 2010), e14284. DOI: 10.1371/journal.pone.0014284.

References

[21] Marco Antoniotti, Alberto Policriti, Nadia Ugel, and Bud Mishra. “Model Building and Model Checking for Biochemical Processes”. In: *Cell Biochemistry and Biophysics* 38 (2003), pp. 271–286.

[22] Murad Banaji. “Cycle structure in SR and DSR graphs: implications for multiple equilibria and stable oscillation in chemical reaction networks”. In: *Transactions on Petri Nets and Other Models of Concurrency V*. Springer-Verlag, 2012, pp. 1–21.

[23] Murad Banaji and Gheorghe Craciun. “Graph-theoretic approaches to injectivity and multiple equilibria in systems of interacting elements”. In: *Communications in Mathematical Sciences* 7.4 (2009), pp. 867–900.

[24] Laurence Calzone, Amélie Gelay, Andrei Zinovyev, François Radvanyi, and Emmanuel Barillot. “A Comprehensive Modular Map of Molecular Interactions in RB/E2F Pathway”. In: *Molecular Systems Biology* 4.173 (2008).

[25] Nathalie Chabrier and François Fages. “Symbolic model checking of biological systems”. In: *Poster proceedings of European Conference on Computational Biology ECCB'02*. Saarbrück, Sept. 2002.

[26] Nathalie Chabrier-Rivier, Marc Chiaverini, Vincent Danos, François Fages, and Vincent Schächter. “Modeling and querying biochemical interaction networks”. In: *Theoretical Computer Science* 325.1 (Sept. 2004). Ed. by Corrado Priami, pp. 25–44.

[27] Bernard Chazelle. “Natural Algorithms and Influence Systems”. In: *Communications of the ACM* 55.12 (Dec. 2012), pp. 101–110. ISSN: 0001-0782. DOI: 10.1145/2380656.2380679.

[28] Patrick Cousot and Radhia Cousot. “Abstract Interpretation: A Unified Lattice Model for Static Analysis of Programs by Construction or Approximation of Fixpoints”. In: *POPL'77: Proceedings of the 6th ACM Symposium on Principles of Programming Languages*. Los Angeles. New York: ACM Press, 1977, pp. 238–252.

[29] Vincent Danos and Cosimo Laneve. “Formal Molecular Biology”. In: *Theoretical Computer Science* 325.1 (2004), pp. 69–110.

[30] Niraj Dudani, Subhasis Ray, Siji George, and Upinder Bhalla. "Multiscale modeling and interoperability in MOOSE". In: *BMC Neuroscience* 10.Suppl 1 (2009), P54. DOI: 10.1186/1471-2202-10-S1-P54.

[31] Steven Eker et al. "Pathway Logic: Symbolic Analysis of Biological Signaling". In: *Proceedings of the seventh Pacific Symposium on Biocomputing*. Jan. 2002, pp. 400–412.

[32] Péter Érdi and János Tóth. *Mathematical Models of Chemical Reactions: Theory and Applications of Deterministic and Stochastic Models*. Nonlinear science : theory and applications. Manchester University Press, 1989. ISBN: 9780719022081.

[33] Martin Feinberg. "Mathematical aspects of mass action kinetics". In: *Chemical Reactor Theory: A Review*. Ed. by L. Lapidus and N. R. Amundson. Prentice-Hall, 1977. Chap. 1, pp. 1–78.

[34] Elisenda Feliu and Carsten Wiuf. "Preclusion of switch behavior in reaction networks with mass-action kinetics". In: *Applied Mathematics and Computation* 219.4 (2012), pp. 1449–1467.

[35] Luis F. de Figueiredo et al. "Computing the shortest elementary flux modes in genome-scale metabolic networks". In: *Bioinformatics* 25.23 (Dec. 2009), pp. 3158–3165. ISSN: 1460-2059. DOI: 10.1093/bioinformatics/btp564.

[36] David Gilbert, Monika Heiner, and Sebastian Lehrack. "A Unifying Framework for Modelling and Analysing Biochemical Pathways Using Petri Nets". In: *CMSB'07: Proceedings of the fifth international conference on Computational Methods in Systems Biology*. Ed. by Muffy Calder and Stephen Gilmore. Vol. 4695. Lecture Notes in Computer Science. Edinburgh, Scotland: Springer-Verlag, Sept. 2007, pp. 200–216. DOI: 10.1007/978-3-540-75140-3_14.

[37] Daniel T. Gillespie. "Exact Stochastic Simulation of Coupled Chemical Reactions". In: *Journal of Physical Chemistry* 81.25 (1977), pp. 2340–2361.

[38] Eva Grafahrend-Belau et al. "Modularization of biochemical networks based on a classification of Petri net by T-invariants". In: *BMC Bioinformatics* 9.90 (Feb. 2008).

[39] Monika Heiner, David Gilbert, and Robin Donaldson. "Petri Nets for Systems and Synthetic Biology". In: *8th Int. School on Formal Methods for the Design of Computer, Communication and Software Systems: Computational Systems Biology SFM'08*. Ed. by M. Bernardo, P. Degano, and G. Zavattaro. Vol. 5016. Lecture Notes in Computer Science. Bertinoro, Italy: Springer-Verlag, Feb. 2008, pp. 215–264.

[40] Reinhart Heinrich and Stefan Schuster. *The Regulation of Cellular Systems*. Springer-Verlag, 1996. ISBN: 9780412032615.

[41] Markus J. Herrgård, Neil Swainston, Paul Dobson, Warwick B Dunn, K Yalçin Arga, et al. "A consensus yeast metabolic network reconstruction obtained from a community approach to systems biology". In: *Nature Biotechnology* 26.10 (Oct. 2008), pp. 1155–1160. DOI: 10.1038/nbt1492.

[42] Michael Hucka et al. "The Systems Biology Markup Language (SBML): A Medium for Representation and Exchange of Biochemical Network Models". In: *Bioinformatics* 19.4 (2003), pp. 524–531.

[43] Hans-Michael Kaltenbach. *A unified view on bipartite species-reaction and interaction graphs for chemical reaction networks*. arXiv:1210.0320. Oct. 2012. eprint: 1210.0320.

[44] Minoru Kanehisa and Susumu Goto. "KEGG: Kyoto Encyclopedia of Genes and Genomes". In: *Nucleic Acids Research* 28.1 (2000), pp. 27–30.

[45] Andrew Kennedy. "Dimension Types". In: *Programming Languages and Systems - Proceedings of ESOP'94, 5th European Symposium on Programming*. Ed. by Donald Sannella. Vol. 788. Lecture Notes in Computer Science. Edinburgh, U.K.: Springer-Verlag, 1994, pp. 348–362. ISBN: 3-540-57880-3. DOI: 10.1007/3-540-57880-3_23.

[46] Kurt W. Kohn. "Molecular Interaction Map of the Mammalian Cell Cycle Control and DNA Repair Systems". In: *Molecular Biology of the Cell* 10.8 (Aug. 1999), pp. 2703–2734.

[47] Kurt W. Kohn. "Molecular interaction maps as information organizers and simulation guides". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 11.1 (2001), pp. 84–97. DOI: 10.1063/1.1338126.

[48] Abdelhalim Larhlimi and Alexander Bockmayr. "A new constraint-based description of the steady-state flux cone of metabolic networks". In: *Discrete Applied Mathematics* 157.10 (2009). Networks in Computational Biology, pp. 2257–2266. ISSN: 0166-218X. DOI: 10.1016/j.dam.2008.06.039.

[49] Arne C. Müller and Alexander Bockmayr. "Flux modules in metabolic networks". In: *Journal of Mathematical Biology* 69.5 (2014), pp. 1151–1179. ISSN: 0303-6812. DOI: 10.1007/s00285-013-0731-1.

[50] Nicolas Le Novère et al. "Minimum information requested in the annotation of biochemical models (MIRIAM)". In: *Nature Biotechnology* 23.12 (Dec. 2005), pp. 1509–1515. DOI: 10.1038/nbt1156.

[51] Kanae Oda, Yukiko Matsuoka, Akira Funahashi, and Hiroaki Kitano. "A comprehensive pathway map of epidermal growth factor receptor signaling". In: *Molecular Systems Biology* 1 (May 2005). DOI: 10.1038/msb4100014.

[52] Andrew Phillips and Luca Cardelli. "A Correct Abstract Machine for the Stochastic Pi-calculus". In: *Transactions on Computational Systems Biology* (2005). Special issue of BioConcur 2004.

[53] Venkatramana N. Reddy, Michael L. Mavrovouniotis, and Michael N. Liebman. "Petri net representations in metabolic pathways". In: *Proceedings of the 1st International Conference on Intelligent Systems for Molecular Biology (ISMB)*. Ed. by Lawrence Hunter, David B. Searls, and Jude W. Shavlik. AAAI Press, 1993, pp. 328–336.

[54] Aviv Regev, William Silverman, and Ehud Y. Shapiro. "Representation and simulation of biochemical processes using the pi-calculus process algebra". In: *Proceedings of the sixth Pacific Symposium of Biocomputing*. 2001, pp. 459–470.

[55] A. Rezola et al. "Exploring metabolic pathways in genome-scale networks via generating flux modes". In: *Bioinformatics* 27.4 (2011), pp. 534–540. DOI: 10.1093/bioinformatics/btq681. eprint: http://bioinformatics.oxfordjournals.org/content/27/4/534.full.pdf+html.

[56] Michael A. Savageau. "Biochemical Systems Analysis II The Steady-state Solutions for an n-pool System using a Power-law Approximation". In: *Journal of Theoretical Biology* 25.3 (1969), pp. 370–379.

[57] Michael A. Savageau and Eberhard O. Voit. "Recasting nonlinear differential equations as S-systems: a canonical nonlinear form". In: *Mathematical Biosciences* 87.1 (1987), pp. 83–115. ISSN: 0025-5564. DOI: 10.1016/0025-5564(87)90035-6.

[58] Paul Shannon, Andrew Markiel, Owen Ozier, Nitin S. Baliga, Jonathan T. Wang, et al. "Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks". In: *Genome Research* 13.11 (2003), pp. 2498–2504. DOI: 10.1101/gr.1239303.

[59] Nicola Soranzo and Claudio Altafini. "ERNEST: a toolbox for chemical reaction network theory". In: *Bioinformatics* 25.21 (2009), pp. 2853–2854. DOI: 10.1093/bioinformatics/btp513.

[60] Christophe Soulé. "Graphic requirements for multistationarity". In: *ComplexUs* 1 (2003), pp. 123–133.

[61] René Thomas, Anne-Marie Gathoye, and Lucie Lambert. "A complex control circuit : regulation of immunity in temperate bacteriophages". In: *European Journal of Biochemistry* 71.1 (Dec. 1976), pp. 211–227.

[62] John J. Tyson. "Modeling the cell division cycle: cdc2 and cyclin interactions". In: *Proceedings of the National Academy of Sciences* 88.16 (Aug. 1991), pp. 7328–7332.

[63] Ionela Zevedei-Oancea and Stefan Schuster. "Topological analysis of metabolic networks based on Petri net theory". In: *In Silico Biology* 3.29 (2003).

[64] Andrei Zinovyev, Eric Viara, Laurence Calzone, and Emmanuel Barillot. "BiNoM: a Cytoscape plugin for manipulating and analyzing biological networks". In: *Bioinformatics* 24.6 (Mar. 2008), pp. 876–877.