Graph matching, theory and SAT implementation

Orianne Bargain

TECHNISCHE UNIVERSITÄT DRESDEN

Faculty of Computer Science International Center for Computational Logic

Diploma Thesis

Graph matching, theory and SAT implementation

Orianne Laura Bargain
Matriculation number: 4672877
Matriculation year: 2017

Supervisors
Dr. Johannes K. Fichte (TU Dresden)
Dr. François Fages (Inria Saclay Ile de France)

Submitted on: 30.09.2019

Statement of authorship

I hereby certify that I have authored this Diploma Thesis entitled *Graph matching, theory and SAT implementation* independently and without undue assistance from third parties. No other than the resources and references indicated in this thesis have been used. I have marked both literal and accordingly adopted quotations as such. There were no additional persons involved in the intellectual preparation of the present thesis. I am aware that violations of this declaration may lead to subsequent withdrawal of the degree.

Dresden, 30.09.2019

Orianne Laura Bargain

Acknowledgements

Foremost, I would like to thank Dr. François Fages, who offered me this diploma thesis subject and who supervised me during this thesis.

I'm grateful to Dr. Johannes Fichte for his support during this project.

I would like to thank Dr. Sylvain Soliman, for sharing with me his expertness of Prolog.

I would like to give credit to Eva Philippe who also worked on subgraph epimorphisms.

My time in Inria Saclay Île-de-France was pleasant thanks to the Lifeware team who welcomed me. All members of this research team were very friendly and helpful.

Finally, I would like to express my gratitude to Inria Saclay Île-de-France, the Technische Universität Dresden and the Institut National des Sciences Appliquées of Rennes, which, all together, made this diploma thesis possible.

Abstract

In systems biology, large biochemical reaction networks can either be represented by bipartite graphs or by systems of ordinary differential equations. Modelers want to determine the existence of reductions between those reaction networks. Because, it is not possible to decide this existence with equation systems, a previous thesis [1] focussed on graph structures. A subgraph epimorphism (SEPI) framework was developed and gave results close to biologists' expectations.

Three main difficulties of the SEPI framework have been identified. First, establishing whether two models are linked through a SEPI is complex and computationally expensive. Second, the number of SEPIs found can be huge, making the analysis of SEPI sets between two given graphs very difficult for biologists. Finally, some existing SEPIs do not have a biological interpretation.

This diploma thesis led to three combined ways to improve the framework. One way consisted to redefine the decision problem into an optimisation problem to restrict the set of solutions. A second way was to determine, together with biologists, restrictions on one of the framework's operations in order to filter irrelevant reductions. Lastly, a preprocessing step has been introduced, consisting of rewriting graphs according to subgraph isomorphism relations. The impact of these three combined implementations has been evaluated on models of the BioModels database. Results demonstrated that it contributed to make the SEPI framework more relevant, efficient and functional.

Contents

1. Introduction

Systems biology relies on computational models. It aims to understand behaviours of complex systems thanks to the analysis and integration of interactions between elementary components. It uses a holistic approach to biological research [2].

For example, systems biology studies molecular interactions in cells and considers the cell as a whole [2]. It uses models to store knowledge. Interactions can be described at different scales, with a variety of quantitative or qualitative approaches. Many formalisms have been developed such as reaction graphs, gene networks and process algebra. Proteins are macromolecules. At their scale, signed graphs, which are graphs with positive or negative labels on each edge, depict the influence of proteins. At a more precise level, signed hypergraphs or signed bipartite graphs represent reaction networks, which indicate which chemical species interact together, by means of a common reaction, to produce other species. With stoichiometry and ordinary differential equations (ODE), simulations can be performed. Simulations give a better understanding of some dynamical behaviour of the system and can highlight existence of stationary states and periodic behaviours.

The Michaelis-Menten enzymatic reaction is a classical example in Systems Biology [3], observed with many enzymes. The reaction can be written in several forms. Michaelis-Menten mechanism is a catalytic mechanism with three reactions and four molecular species in its expanded form: enzyme E, substrate S, complex enzyme-substrate C, also denoted by ES, product P. The mechanism describes kinetics of an enzyme-catalysed reaction acting on a single substrate to irreversibly give a product. Its reaction model and mass action law kinetics is

$$E + S \underset{k_c}{\overset{k_d}{\leftrightarrows}} C \rightarrow_{k_p} E + P$$

which gives the following system of ODEs where capital letters denote the concentration of corresponding species:

$$\begin{cases} \dot{S} &= -k_c \cdot E \cdot S + k_d \cdot C \\ \dot{E} &= -k_c \cdot E \cdot S + k_d \cdot C + k_p \cdot C \\ \dot{C} &= k_c \cdot E \cdot S - k_d \cdot C - k_p \cdot C \\ \dot{P} &= k_p \cdot C \end{cases}$$

Michaelis-Menten mechanism can be reduced to a single reaction with three species (E, S, P) and more complex kinetics:

$$E + S \rightarrow_{\left[\frac{k_m \cdot K \cdot S}{k_m + S}\right]} E + P,$$

with system of ODEs:

$$\begin{cases} \dot{S} &= -\frac{k_m \cdot K \cdot S}{k_m + S} \\ \dot{E} &= 0 \\ \dot{P} &= \frac{k_m \cdot K \cdot S}{k_m + S} \end{cases}$$

These two mechanisms will sometimes be referred to as expanded and reduced forms of Michaelis-Menten reaction.

This reduction of the Michaelis-Menten mechanism is valid for some initial concentrations of molecular species. Figures 1-1 and 1-2 show the dynamic behaviour of species concentrations in expanded and reduced Michaelis-Menten mechanisms. Dynamic behaviours of species of interest (E, S, P) is almost the same in both mechanisms, which shows the relevance of this reduction.

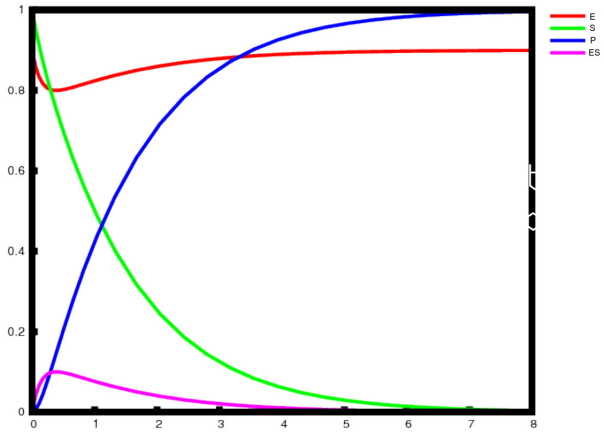

Figure 1-1.: Evolution of concentrations: model with three reactions.

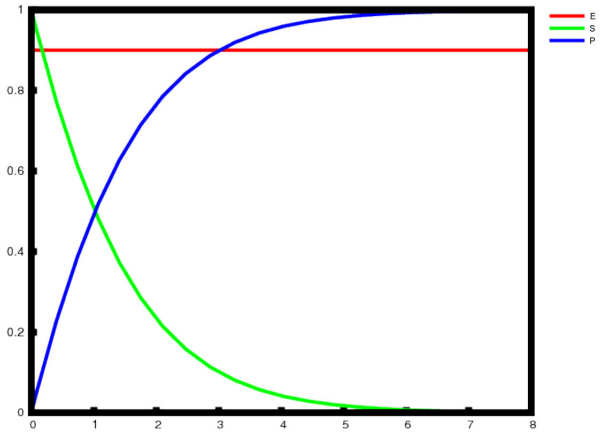

Figure 1-2.: Evolution of concentrations: model with one reaction.

A free open-source repository called BioModels [4] was created for storing, exchanging, and retrieving quantitative models of biological interest, which were described in peer-reviewed scientific literature.

But mathematical biology is not suited for reduction on complex reactions. Similarities of mechanism are better revealed by relationships between network structures. Thus, current approaches focus on reaction graphs by making abstraction of stoichiometry and kinetic.

Figures 1-3 and 1-4 are examples of reaction graphs and they show a same mechanism. Reaction graphs are bipartite graphs that represent a biochemical reaction. Species are depicted by round vertices and reactions are depicted by square vertices. An edge represents a link in a reaction. For example, specie E and specie S are reacting together through reaction R_1 to produce species ES.

Models can have a lot of vertices and can be complex, for example, with more than three hundred vertices on a model representing dynamics of extracellular-signal-regulated kinase [5]. Thus, biologists need automated tools to exhibit reduction relations between models. Especially for big repositories like BioModels [4].

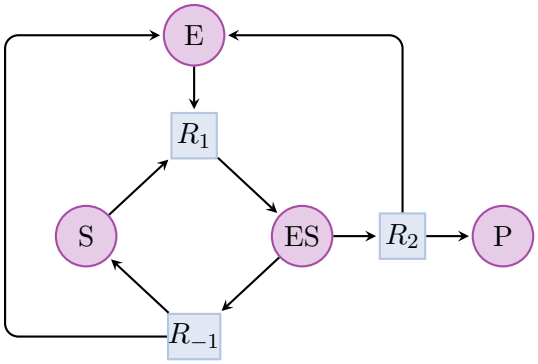

Figure 1-3.: Complete Michaelis-Menten reaction graph.

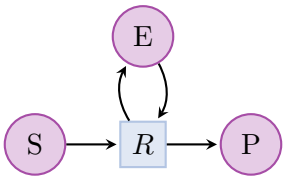

Figure 1-4.: Reduced Michaelis-Menten reaction graph.

1.1. Related work

A few current approaches are reducing biochemical reactions using reaction networks, subgraph matching tools or subgraph epimorphisms.

Subgraph isomorphism

A lot of research has been carried out on the subgraph isomorphism (SISO) problem. GraphGrep [6] and Grafil [7] for example, are graph matching methods using subgraph isomorphism but it gives limited results on reaction graphs.

A common example for limitations of SISO is Michaelis-Menten reaction graph. Michaelis-Menten kinetic is one of the best-known models of enzyme kinetics in biochemistry [3]. Its reaction is presented in figure 1-3. Figure 1-4 is a reduced way to represent Michaelis-Menten kinetic. SISO relation can be seen as a sequence of delete operations performed on vertices. This relation is too strong because it doesn't find a relation between both graphs. Here, a way to reach the second reaction graph from the first one is to delete reaction R_{-1}, merge together reactions R_1 and R_2 and delete specie ES.

Graph minor

Graph minors [8, 9] is another well studied graph theory concept. It has a lot of properties but because reaction graphs are bipartite graphs, no solution to extend graph minor definition to biochemical models have been found.

Substructure index-based approximate graph alignment

Substructure Index-Based Approximate Graph Alignment (SAGA) tool [10] was developed to overcome SISO limitations. SAGA is an approximate subgraph matching tool, which permits node gaps, node mismatches, and graph structural differences. A node gap is a node insertion or deletion, node mismatches exhibit similar functionality of specie nodes and graph structural differences allow flexibility in node connectivity relationships.

But SAGA was implemented to query databases and not to compare two given graphs.

Morphisms of reaction networks

Another approach [11] uses influence networks to represent complex biological systems. Influence networks depict the impact of molecular species on each others. Cardelli studies structural aspects of models to infer properties and makes a connection between network structure and behaviour.

Figures 1-5 and 1-6 are examples of influence networks. They are an abstraction for more detailed biochemical interactions. Each node represents an influence species. A ball-head represents an activation influence, a bar-head represents an inhibition influence, and a simple edge represents an outgoing influence to another node.

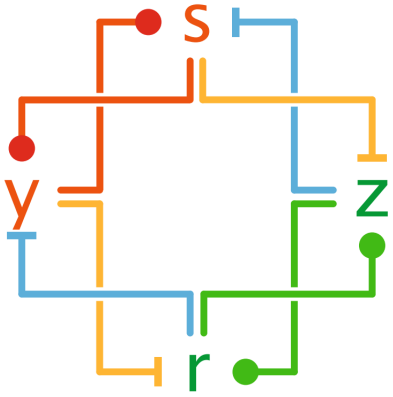

Figure 1-5.: Complex influence network.

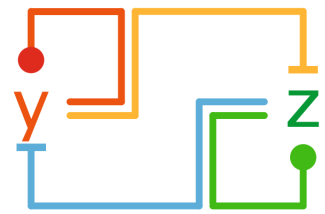

Figure 1-6.: Simplified influence network.

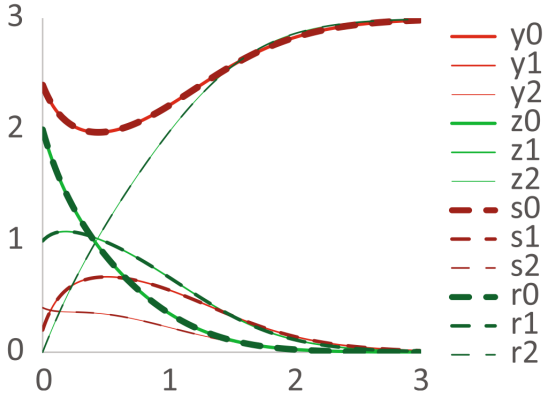

Figure 1-7.: Concentration of species over time of complex network.

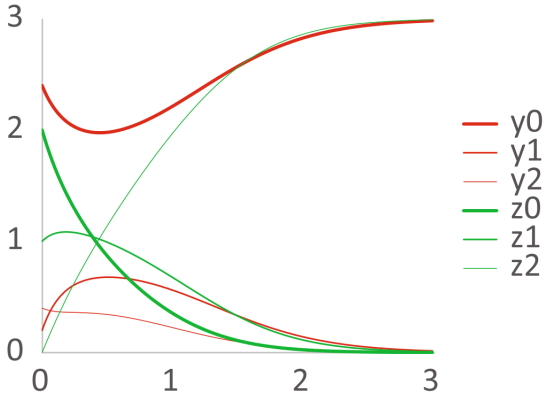

Figure 1-8.: Concentration of species over time of simplified network.

Figure 1-6 is a reduction of figure 1-5. Figure 1-8 and figure 1-7 are corresponding concentration graphs.

Cardelli defines morphisms between reaction networks with structural connections. It achieved to determine statically that a complex network imitate a simpler one. Figure 1-5 to figure 1-8 are examples. Simplified influence network reproduce kinetics of complex influence network with same initial conditions and reaction rates.

The study [11] shows that graph morphisms provide a structural reason for kinetic similarity, but it doesn't bring a tool to automatically compare biochemical reactions.

Subgraph epimorphism

SISO problem can be viewed as a sequence of delete operations. To tackle SISO's limitations, a solution is to add a merge operation to the delete operation, which is equivalent to a subgraph epimorphism (SEPI) problem. A previous thesis [1] introduced a SEPI framework[1] for model reductions in systems biology, establishing a formal relation between models.

SEPI framework focuses only on reaction graphs but corresponds to a mathematical reduction, since a vertex deletion can be viewed as neglecting a specie and a fusion of two vertices as an aggregation of two species. To obtain one graph from another, a sequence of vertex deletion and fusion is applied. These two simple rules are accurate enough to capture mathematical model reductions and not too complex. When a model is reachable from another model with this framework, it means that these models might be reduced in some valid way and it give insights as of how much a model is similar to another. SEPI is a powerful framework because, without having to add more information to models, it can organise existing models as hierarchies of refined to reduced. Comparison between reaction graphs can be applied systematically to repositories such as BioModels that contain hundreds of models with no given relation.

1.2. Motivations

Three main difficulties of SEPI framework have been identified by biologists when using it. First, establishing whether two models are linked through a SEPI is complex and computationally expensive. Second, the number of SEPIs found can be huge, making analysis of SEPI sets between two given graphs very difficult for biologists. Finally, some existing SEPIs do not have a biological interpretation.

Additionally, there is no guarantee that SEPI pairings are related to correct reductions. Up to now, there is no perfect correspondence between valid model reductions and graph operations, partly because valid model reductions themselves are very difficult to justify mathematically. For example, L. Noethen et al. [12] uses Tikhonov [13] and Fenichel [14] theorems.

The goal of this diploma thesis is to study further model reductions and to improve SEPI framework. This work uses and refines the graph theoretical framework of subgraph epimorphisms developed by S. Gay, F. Fages, and S. Soliman in [1, 15, 16].

[1] SEPI framework was implemented in an open-source software modelling platform called Biocham, available at `https://lifeware.inria.fr/biocham4`.

1.3. Contributions

This diploma thesis led to three combined ways to improve the framework.

1. One way consists to redefine a decision problem into an optimisation problem to select solutions.
2. A second way is to determine, together with biologists, restrictions on one of the framework's operations in order to filter irrelevant reductions.
3. Lastly, a preprocessing step is introduced, consisting of rewriting graphs according to subgraph isomorphism relations.

Impacts of these three combined implementations are evaluated on models of BioModels database. Results demonstrate that it contributed to make SEPI framework more relevant, efficient, and functional.

1.4. Outline

Approaches used and improved in this thesis have specific notations and uses specific tools, thus Chapter 2 provides context and explanations to readers not familiar with graph theory or SAT solving. Chapter 3 presents a first approach enhancing SEPI framework: the decision problem is redefined into an optimisation problem by adding an objective of maximisation or minimisation of the vertex deletion number. A second approach determines restrictions on a merge operation in order to filter irrelevant reductions, it is demonstrated in Chapter 4. A more functional variant of this merge restriction is explained and implemented in Chapter 5. Chapter 6 introduces a preprocessing step consisting of rewriting graphs according to subgraph isomorphism relations. Chapter 7 tackles the problem of finding a Greatest Lower Bound (GLB) or a Lowest Upper Bound (LUB) between two graphs that are not linked by a SEPI relation. Bounds correspond to a common reduction or a common refinement of both graphs. Finally, Chapter 8 concludes this thesis and discuss some axes of enhancement.

2. Preliminaries

This chapter assumes familiarity with basic set theory and provides background knowledge for readers who are not familiar with graph theory and SAT solving.

2.1. Graph theory

Basics of graph theory will not be presented in this chapter. Readers can refer to the work of Diestel [17] or Bang-Jensen and Gutin [18].

Definition 2.1.1 (Directed graph) *A directed graph G is a pair $G = (V, A)$, where V is a set of vertices and $A \subseteq V \times V$ is a set of arcs.*

When it is not specified, *graph* means *directed graph*.

A graph may have loops and cannot have more than one arc from some vertex to another one. The cardinality of a set S is denoted $|S|$. When not explicitly defined, G and G' denote graphs, with $G = (V, A)$ and $G' = (V', A')$.

Operations

The subgraph epimorphism will be based on a merge and a delete operation.

A delete operation removes a vertex and all arcs connected to it.

Definition 2.1.2 (Deletion) *Let $u \in V$. The result of the delete operation d_u is graph $d_u(G) = (V', A')$ where:*

$$V' = V \setminus \{u\} \text{ and}$$
$$A' = A \cap (V' \times V').$$

$G \to_d G'$ is written when there exists $u \in V$ such that $d_u(G) = G'$. $G \to_d^* G'$ is written when G' can be obtained from G using zero, one or several deletes.

A merge operation removes both initial vertices and creates a new corresponding vertex that inherits arcs connected to them.

Definition 2.1.3 (Merge) *For all $u, v \in V$, the result of the merge operation $m_{u,v}$ is graph $m_{u,v}(G) = (V', A')$ where:*

$$V' = V \setminus \{u, v\} \uplus \{uv\},$$
$$A' = A \cap (V' \times V'),$$
$$\cup\{(uv, x) | (u, x) \in A\} \cup\{(uv, x) | (v, x) \in A\},$$
$$\cup\{(x, uv) | (x, u) \in A\} \cup\{(x, uv) | (x, v) \in A\},$$
$$\cup\{(uv, uv) | (u, v) \in A\} \cup\{(uv, uv) | (v, u) \in A\}.$$

$G \to_m G'$ is written when there exists u and v such that $m_{u,v}(G) = G'$. $G \to_m^* G'$ is written when G' can be obtained from G using zero, one, or several merges.

$G \to_{md} G'$ is written when $G \to_m G'$ or $G \to_d G'$. $G \to_{md}^* G'$ is written when G' can be obtained from G using zero, one, or several merges and/or deletes.

Merge and delete operations are actions on a graph structure, but they can be biologically construed. Some possible interpretations are:

- Deletion of a specie.
  Specie's concentration is constant or its variation is insignificant compared to the absolute value. The specie can be ignored.
- Deletion of a reaction.
  Reaction's rate is zero or is deemed negligible compared to another.
- Merging of two species.
  Two species have proportional concentrations and can be confused.
- Merging of two reactions.
  Two reactions have proportional rates.

Morphisms

From the delete and the merge operation definition, subgraph epimorphism can be defined.

Definition 2.1.4 (Subgraph Epimorphism) *A subgraph epimorphism (SEPI) [1] from G to G' is a function $\mu : V \to V' \cup \{\bot\}$ such that :*

$\forall (u, v)$ *in* $\mu^{-1}(V')$, $(u, v) \in A \implies (\mu(u), \mu(v)) \in A'$,

μ *surjective on* V' *and* A'.

For reaction graphs, a SEPI verifies $\mu(S) \subseteq S'$ and $\mu(R) \subseteq R'$.

Theorem 2.1.1 *[1] The existence of a SEPI from G to G' is equivalent to the existence of a finite sequence of merge and delete operations that yields a graph isomorphic to G' when applied to G.*

$G \overset{SEPI}{\rightsquigarrow} G'$ is written when there exists a SEPI from G to G'.

Definition 2.1.5 (SEPI decision problem) *Subgraph epimorphism problem is the decision problem:*

Instance: Two Graphs G, G'.

Question: $G \overset{SEPI}{\rightsquigarrow} G'$?

This chapter assumes familiarities with basics of theoretical computer science. Otherwise readers can refer to the work of Van Leeuwen [19].

Theorem 2.1.2 *[1] Subgraph epimorphism problem is NP-complete.*

Partial order

Definition 2.1.6 (Total order) *[20] A relation $\leq$ is a total order on a set S if the following properties hold.*

- *Reflexivity: $a \leq a$ for all $a \in S$.*
- *Antisymmetry: $a \leq b$ and $b \leq a$ implies $a = b$.*
- *Transitivity: $a \leq b$ and $b \leq c$ implies a leqc.*
- *Comparability: for any $a, b \in S$, either $a \leq b$ or $b \leq a$.*

A total order on graphs can be defined. $G \leq G'$ implies $\mid V \mid \leq \mid G' \mid$.

Definition 2.1.7 (Partial order) *[20] A relation $\leq$ is a partial order on a set S if the following properties hold.*

- *Reflexivity: $a \leq a$ for all $a \in S$.*
- *Antisymmetry: $a \leq b$ and $b \leq a$ implies $a = b$.*
- *Transitivity: $a \leq b$ and $b \leq c$ implies a leqc.*

A partial order can be defined on a set of graphs.

Definition 2.1.8 (SEPI partial order) *Let G, G' be two graphs, $G' \leq G$ is defined if and only if $G \overset{SEPI}{\rightsquigarrow} G'$.*

Definition 2.1.9 (Set of lower bounds) *A set of lower bounds can be defined by $G \cap_{md} G' = \{H | G \rightarrow^*_{md} H \wedge G' \rightarrow^*_{md} H\}$.*

A *maximal* element of a set X is an element $x \in X$ such that for all $y \in X, y < x$. x is the *maximum* of X if it is unique.

Definition 2.1.10 (Set of greatest lower bounds (glb)) *$\overline{G \cap_{md} G'}$ is a set of $\rightarrow^*_{md}$-maximal elements of $G \cap_{md} G'$.*

Definition 2.1.11 (Set of upper bounds) *A set of upper bounds can be defined by $G \cup_{md} G' = \{H | H \rightarrow^*_{md} G \wedge H \rightarrow^*_{md} G'\}$.*

A *minimal* element of a set X is an element $x \in X$ such that for all $y \in X, x < y$. x is the *minimum* of X if it is unique.

Definition 2.1.12 (Set of least upper bounds (lub)) *$\underline{G \cup_{md} G'}$ is a set of $\rightarrow^*_{md}$-minimal elements of $G \cup_{md} G'$.*

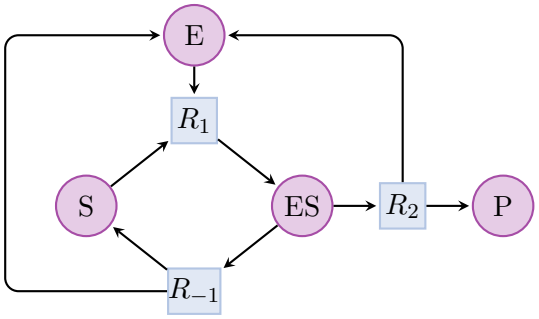

Figure 2-1.: Complete Michaelis-Menten reaction graph

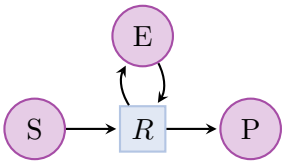

Figure 2-2.: Reduced Michaelis-Menten reaction graph

2.2. Reaction graph

Definition 2.2.1 (Reaction graph) *A reaction graph G is written $G = (S \cup R, A)$ such that S is a set of specie nodes, R is a set of reaction nodes, and $A \subseteq S \times R \cup R \times S$ is a set of arcs that describes how species interact through reactions.*

Example In figure 2-1, the graph G is defined as $G = (S, R, A)$ with $S = \{E, S, ES, P\}$, $R = \{R_1, R_{-1}, R_2\}$, and $A = \{(E, R_1), (S, R_1), (R_1, ES), (ES, R_{-1}), (R_{-1}, E), (R_{-1}, E), (ES, R_2), (R_2, E), (R_2, P)\}$.

In figure 2-2, the graph G' is defined as $G' = (S', R', A')$ with $S' = \{E, S, P\}$, $R' = \{R\}$ and $A' = \{(E, R), (S, R), (R, E), (R, P)\}$.

There is a SEPI from G to G' by deleting R_{-1}, merging R_1 with R_2 and merging E with ES.

2.3. Boolean satisfiability problem

In previous works [1], subgraph epimorphism problem has been encoded into a SAT problem.

Problem statement

A *Boolean variable* is a variable that can either be true (represented by a value 1) or false (represented by a value 0). Let X be a set of Boolean variables. A *literal* l is either x or its negation (denoted $\bar{x}$), with $x \in X$. A *clause* is a disjunction (denoted $\vee$) of literals. A formula in conjunctive normal form (or *CNF formula*) is a conjunction (denoted $\wedge$) of clauses. An *assignment* of a variable is to give a value 0 or 1 to the variable. An *interpretation* of a CNF formula is an assignment of its variables. A formula is said to be *satisfiable* if there exists an interpretation that makes the formula true. An interpretation *satisfies* a formula if it makes the formula true. A formula is said to be *unsatisfiable* if the formula is false for all possible interpretations.

Definition 2.3.1 (SAT problem) *A Boolean satisfiability problem (SAT for short) is the problem of determining whether there exists an interpretation that satisfies a given CNF formula.*

Problem resolution

A naive method to solve a SAT problem is to construct a truth table but the number of assignments is exponential in the number of variables of the input formula.

A systematic method with a binary search tree is more efficient. To prove the satisfiability of a CNF formula Φ, a literal l can be assigned, then, satisfiability of $l \vee \Phi$ or $\neg l \vee \Phi$ is recursively proved.

DPLL algorithm [21, 22] is based on this idea and on two other rules (*unit* and *pure*) that reduce the search space.

Modern SAT solvers are based Conflict-Driven Clause Learning (CDCL) algorithms [23] which are inspired by DPLL algorithm, backtracking by conflict analysis with clause recording [24], and Boolean constraint propagation using watched literals [25].

Those tools are very efficient in practice while the problem is very hard in theory [26]. In worst case scenario, current SAT-solving algorithms must take exponential time to solve a problem but on the other hand solvers are applied successfully on large industrial instances.

Solver input

Input of a SAT solver is a text file in a simplified version of DIMACS format [27]. Besides comments, first line of a file should be *p cnf nb_variables nb_clauses*, where *nb_variables* is an upper bound on the largest index of a variable appearing in the file or the exact number depending on the implementation, and *nb_clauses* is the exact number of clauses.

Other lines of a file represent a CNF formula. Each variable corresponds to an integer, its negation is depicted by the inverse of corresponding integer. Each line is a disjunction of literals and stand for a clause. Each line end with a zero. The CNF formula is a conjunction of all clauses.

An example of input file:

```
c comments
p cnf 5 3
1 -5 4 0
-1 5 3 4 0
-3 -4 0
```

2.4. Maximum satisfiability problem

Maximum satisfiability problem [28] is a generalisation of Boolean satisfiability problem where a CNF formula can be unsatisfiable.

Definition 2.4.1 (MAX-SAT problem) *A Maximum Satisfiability problem, MAX-SAT problem for short, is the problem of determining a maximum number of clauses for which there exists an interpretation that satisfies a given CNF formula.*

A weighted clause is a pair (C_i, w_i), where C_i is a clause and w_i a positive integer representing its weight. A weighted CNF formula, WCNF formula for short, is a set of weighted clauses. The weighted version of MAX-SAT problem is to find an interpretation that maximise the combined weight of the satisfied clauses. Hard clauses are clauses that must be satisfied, soft clauses are non-mandatory clauses. A cost of an interpretation is the sum of weights of clauses that are unsatisfied.

Input of a weighted MAX-SAT solver is also a text file in a simplified version of DIMACS format. Besides comments, first line of a file should be *p cnf nb_variables nb_clauses top*, where *nb_variables* is an upper bound on the largest index of a variable appearing in the file or the exact number, *nb_clauses* is the exact number of clauses and top is an integer representing the maximum weight. Hard clauses have weight top and soft clauses have a weight smaller than top.

An example of MAX-SAT solver input file:

```
1  c comments
2  p wcnf 4 5 16
3  16 1 -2 4 0
4  16 -1 -2 3 0
5  8 -2 -4 0
6  4 -3 2 0
7  3 1 3 0
```

2.5. Previous implementation

Subgraph epimorphism problem has been encoded into instances of SAT by S. Gay [1] and then resolved by a SAT solver. To describe a SAT instance, particular notations will be used.

2.5.1. Notations

Variables in bold font are SAT variables. *cl(...)* represents a clause. $m(u)$ represents the image of u through morphism m. $\mathbf{m}_{u,u'} = 1$ if and only if $m(u) = u'$.

Elements of $V' \cup \bot$ are put in a total order $v'_0 = \bot < v'_1 < ... < v'_{n'}$.

$\mathbf{m}^{<}_{u,u'} = 1$ if and only if $m(u) < u'$. $m((u,v)) = (u',v')$ represents the image of (u,v) through morphism m. $\mathbf{m}_{(u,v),(u',v')} = 1$ if and only if $m((u,v)) = (u',v')$. $\mathbf{deleted}_{(u,v)} = 1$ if and only if $m((u,v)) = \bot$. $type(u)$ represents the type of a vertex u (specie or reaction).

2.5.2. Partial surjective function coding

First, partial surjective property of SEPI relation need to be encoded.

Left totality clauses are ensuring that each vertex of a starting graph has at least one image in a targeted graph or its image is bottom.

Right totality clauses are ensuring that each vertex of a targeted graph has at least one antecedent in a starting graph.

Functionality clauses are ensuring that each vertex of a starting graph has only one image through the morphism.

Clauses

I Left totality. $F_{l-tot} := \bigwedge_{\forall u \in S \cup R} cl(\bigvee_{u' \in S' \cup R' \cup \bot} \mathbf{m}_{u,u'})$.

Number of clauses: $|S| + |R|$, number of new variables: $(|S| + |R|) \times (1 + |S'| + |R'|)$.

II Right totality. $F_{r-tot} := \bigwedge_{\forall u' \in S' \cup R'} cl(\bigvee_{u \in S \cup R} \mathbf{m}_{u,u'})$.

Number of clauses: $|S'| + |R'|$, number of new variables: 0.

III Functionality.

i $F_{func-i} := \bigwedge_{\forall (u,u'_j) \in (S \cup R) \times (S' \cup R' \cup \bot)} cl(\mathbf{m}_{u,u'_j} \implies \mathbf{m}^{<}_{u,u'_{j+1}})$,

ii $F_{func-ii} := \bigwedge_{\forall (u,u'_j) \in (S \cup R) \times (S' \cup R' \cup \bot)} cl(\mathbf{m}^{<}_{u,u'_j} \implies \mathbf{m}^{<}_{u,u'_{j+1}})$,

iii $F_{func-iii} := \bigwedge_{\forall (u,u'_j) \in (S \cup R) \times (S' \cup R' \cup \bot)} cl(\mathbf{m}^{<}_{u,u'_j} \implies \neg \mathbf{m}_{u,u'_{j+1}})$.

Number of clauses: $3 \times (|S| + |R|) \times ((|S'| + |R'|) + 1)$, number of new variables: $(|S| + |R|) \times (1 + |S'| + |R'|)$.

2.5.3. Subgraph epimorphism coding

Subgraph epimorphism property is implemented as follow.

Left totality on arcs clauses are ensuring that each edge of a starting graph has an image in a targeted graph or is deleted.

Right totality on arcs clauses are ensuring that each edge of a targeted graph has an antecedent in a starting graph.

Graph morphism clauses are linking SAT variables for edges and SAT variable for vertex by forcing endpoints of starting graph edges (respectively targeted graph edges) to have the corresponding image (respectively antecedent) through morphism m.

Subgraph morphism clauses are ensuring that if a vertex u is deleted, each edge with u as endpoint is also deleted. They also ensure that if an edge is deleted, at least one of its endpoint is deleted.

Redundant morphism propagation clauses are ensuring that the image of a starting graph edge is an edge in a targeted graph.

Bi-graph constraints clauses are ensuring that the image of a starting graph vertex through a morphism m is a vertex of same type in the targeted graph. Type being species or reaction.

Clauses

I Left totality on arcs. $F_{l-tot-arcs} := \bigwedge_{\forall a \in A} cl(\mathbf{deleted}_a \vee \bigvee_{a' \in A'} \mathbf{m}_{a,a'})$.

Number of clauses: $|A|$, number of new variables: $|A| \times (|A'| + 1)$.

II Right totality on arcs. $F_{r-tot-arcs} := \bigwedge_{\forall a' \in A'} cl(\bigvee_{a \in A} \mathbf{m}_{a,a'})$.

Number of clauses: $|A'|$, number of new variables: 0.

III Graph morphism.

i $F_{morph-i} := \bigwedge_{\forall (u,v) \in A, (u',v') \in A'} cl(\mathbf{m}_{(u,v),(u',v')} \implies \mathbf{m}_{u,u'})$,

ii $F_{morph-ii} := \bigwedge_{\forall (u,v) \in A, (u',v') \in A'} cl(\mathbf{m}_{(u,v),(u',v')} \implies \mathbf{m}_{v,v'})$,

iii $F_{morph-iii} := \bigwedge_{\forall (u,v) \in A, (u',v') \in A'} cl((\mathbf{m}_{u,u'} \wedge \mathbf{m}_{v,v'}) \implies \mathbf{m}_{(u,v),(u',v')})$.

Number of clauses: $3 \times |A| \times |A'|$, number of new variables: 0.

IV Subgraph morphism.

i $F_{sub-morph-i} := \bigwedge_{\forall (u,v) \in A} cl(\mathbf{deleted}_{(u,v)}) \implies \mathbf{m}_{u,\perp} \vee \mathbf{m}_{v,\perp})$,

ii $F_{sub-morph-ii} := \bigwedge_{\forall (u,v) \in A} cl(\mathbf{m}_{u,\perp} \implies \mathbf{deleted}_{(u,v)}))$,

iii $F_{sub-morph-iii} := \bigwedge_{\forall (u,v) \in A} cl(\mathbf{m}_{v,\perp} \implies \mathbf{deleted}_{(u,v)}))$.

Number of clauses: $3 \times |A|$, number of new variables: 0.

V Redundant morphism propagation.

$F_{morph-prop} := \bigwedge_{\substack{\forall (u,v) \in A, \\ (u',v') \in ((S' \cup R') \times (S' \cup R')) \setminus A'}} cl(\neg \mathbf{m}_{u,u'} \vee \neg \mathbf{m}_{v,v'})$.

Number of clauses: $|A| \times ((|S'| + |R'|)^2 - |A'|)$, number of new variables: 0.

VI Bi-graph constraint. $F_{bi-graph} := \bigwedge_{\substack{\forall (u,u') \in (S \cup R) \times (S' \cup R'), \\ type(u) \neq type(u')}} cl(\neg \mathbf{m}_{u,u'}))$.

Number of clauses: $|R|^2$, number of new variables: 0.

The formula is then defined as $F_{SEPI} := F_{l-tot} \wedge F_{r-tot} \wedge F_{func-i} \wedge F_{func-ii} \wedge F_{func-iii} \wedge F_{l-tot-arcs} \wedge F_{r-tot-arcs} \wedge F_{morph-i} \wedge F_{morph-ii} \wedge F_{morph-iii} \wedge F_{sub-morph-i} \wedge F_{sub-morph-ii} \wedge F_{sub-morph-iii} \wedge F_{morph-prop} \wedge F_{bi-graph}$.

3. Constraints on the number of deletions

This chapter presents a first approach to reduce the number of SEPI relations between two given graphs. When it is not specified, a SEPI relation, or SEPI, is a sequence of merge and delete operations that relate a graph to another. In a first section, goals are reminded. In a second section, a new approach is presented. An evaluation on BioModels database is shown in a third section. This chapter ends with a brief discussion and a conclusion.

3.1. Motivations

A major drawback of SEPI framework is the amount of SEPI found between reaction graphs. Some existing SEPIs do not have a biological interpretation and the set of SEPIs between two given graphs is often too big to be studied and to be understood. One of targeted goals is to reduce the number of distinct SEPIs between two given graphs.

Figure 3-1 is an example of classes of models from BioModels database. Each vertex represents a model and an arrow between two graphs means that there exists a SEPI relation between those two models. When there is a SEPI from a graph G_1 to a graph G_2 another one from G_2 to G_3 and a last one from G_1 to G_3, arrow corresponding to the last SEPI is not drawn, in other words, transitive closure is not represented in this figure. Numbers on an arrows are numbers of different SEPI existing between two given graphs which is the number of different merge and delete operations that can relate a graph to another. When a number is equal to 200 it means that there is 200 or more different SEPI relations. This size is already too big to be studied and it can exists thousands of relations, all of them were not computed.

Figure 3-1 shows the existence of a lot of SEPI sets with a size greater than 200 and this is not optimal. $\mathcal{S}(G, G')$ will represents a set of SEPI pairings from a graph G to a graph G'.

Figure 3-2 and figure 3-3 are an example on small graphs of an explosion of $\mathcal{S}(G, G')$ size. Figure 3-2 is a model created by M. Marhl et al. [29]. It represents complex calcium oscillations and mitochondria and cytosolic proteins roles in cells. Figure 3-3 is a model studied by J. M. Borghans et al. [30]. It also represents complex intracellular calcium oscillations. Both graphs are from Calcium Oscillations class.

On both figures, circle vertices represent species and square vertices represent reactions. An arrow is drawn from species to a reaction when they are reactants of the reaction. An arrow is drawn from reactions to a specie when the specie is a product of the reactions.

Table 3-1 presents all possible SEPIs from initial graph of figure 3-2 to targeted graph of figure 3-3. Each line represents a SEPI. Each column represents an initial graph's vertex and for each SEPI it's image in the targeted graph is presented. $\perp$ represents a vertex deletion.

It can be observed in table 3-1 that, even on a small example, there are 16 solutions but only 4 that minimise the number of deletions. It is interesting for biologists to see only SEPIs that minimise or maximise the number of vertex deletion. Minimising indicates what has to be deleted imperatively and where vertices can be merged. Maximising indicates all vertices that can be deleted and makes a SEPI more readable.

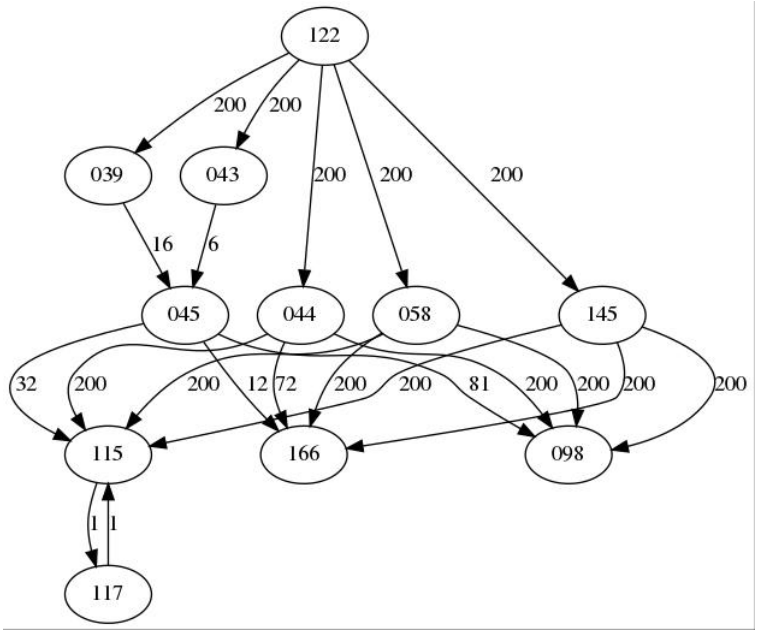

Figure 3-1.: Number of SEPIs between models in Calcium Oscillations class.

This example also raises a problem of symmetries: parts of a reaction graph can have a same structure, the only difference being specie and reaction names it is also called an automorphism. A lot of them can be observed in table 3-1. Symmetries in initial graph G or image graph G' will have a multiplicative effect on the number of SEPIs: if μ is a SEPI from G to G', then for any σ automorphism of G and σ' automorphism of G', $\sigma' \circ \mu \circ \sigma$ is also a SEPI from G to G'.

In Michaelis-Menten example as well as in graphs of figure 3-2 and figure 3-3 described previously, a source of combinatorics is vertices that can indifferently be deleted or merged. It motivates a definition of a partial order on sets $\mathcal{S}(G, G')$ that can be described more compactly by exhibiting only maximal or minimal elements. These extremal elements are sufficient to capture relevant information on relation between G and G', for example to get a set of possible images (respectively pre-images) of each species vertex in G (respectively G').

However, there is no practical way to isolate these extremal elements. Isolating pairings that make extremal the number of deleted vertices (denoted min $\perp$ and max $\perp$) is another solution. Goals of minimizing or maximizing the number of deleted vertices are heuristics, they highlight SEPI pairings that have a chance to be relevant.

3.2. Implementation

To select solutions according to their number of vertex deletions, the SEPI decision problem can be redefined into an optimisation problem. First a few definitions are presented, then clauses are introduced.

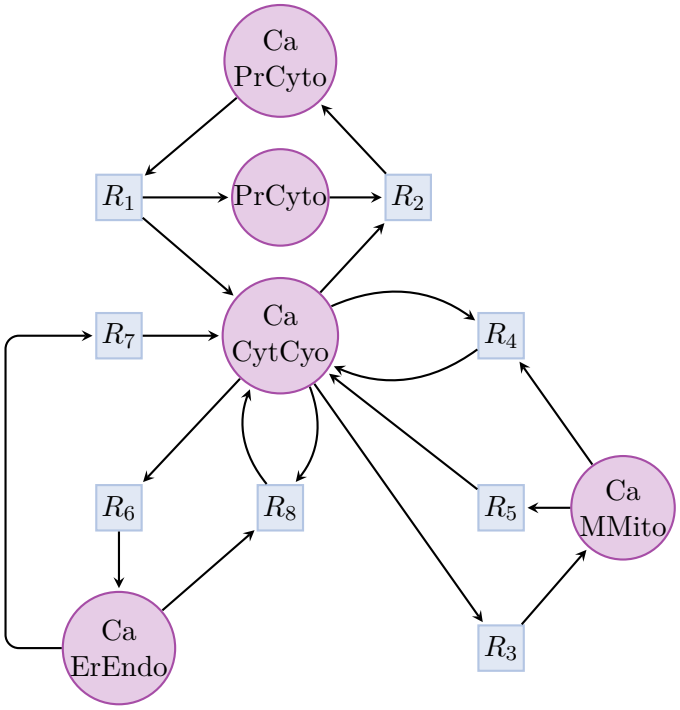

Figure 3-2.: Calcium Oscillations: initial graph.

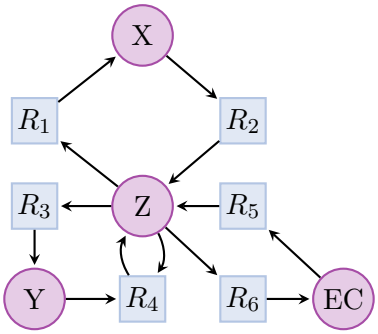

Figure 3-3.: Calcium Oscillations: targeted graph.

	CaPrCyto	Ca_cytCyo	PrCyto	CaMMito	CaErEndo	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	$\perp$
1	EC	Z	Z	X	Y	R_5	R_6	R_1	$\perp$	R_2	R_3	R_4	R_4	1
2	X	Z	Z	EC	Y	R_2	R_1	R_6	$\perp$	R_5	R_3	R_4	R_4	1
3	EC	Z	Z	Y	X	R_5	R_6	R_3	R_4	R_4	R_1	R_2	$\perp$	1
4	X	Z	Z	Y	EC	R_2	R_1	R_3	R_4	R_4	R_6	R_5	$\perp$	1
5	EC	Z	Z	X	Y	R_5	R_6	R_1	$\perp$	R_2	R_3	$\perp$	R_4	2
6	X	Z	Z	EC	Y	R_2	R_1	R_6	$\perp$	R_5	R_3	$\perp$	R_4	2
7	EC	Z	Z	Y	X	R_5	R_6	R_3	R_4	$\perp$	R_1	R_2	$\perp$	2
8	X	Z	Z	Y	EC	R_2	R_1	R_3	R_4	$\perp$	R_6	R_5	$\perp$	2
9	X	Z	$\perp$	EC	Y	R_2	R_1	R_6	$\perp$	R_5	R_3	R_4	R_4	2
10	EC	Z	$\perp$	X	Y	R_5	R_6	R_1	$\perp$	R_2	R_3	R_4	R_4	2
11	EC	Z	$\perp$	Y	X	R_5	R_6	R_3	R_4	R_4	R_1	R_2	$\perp$	2
12	X	Z	$\perp$	Y	EC	R_2	R_1	R_3	R_4	R_4	R_6	R_5	$\perp$	2
13	EC	Z	$\perp$	X	Y	R_5	R_6	R_1	$\perp$	R_2	R_3	$\perp$	R_4	3
14	X	Z	$\perp$	EC	Y	R_2	R_1	R_6	$\perp$	R_5	R_3	$\perp$	R_4	3
15	EC	Z	$\perp$	Y	X	R_5	R_6	R_3	R_4	$\perp$	R_1	R_2	$\perp$	3
16	X	Z	$\perp$	Y	EC	R_2	R_1	R_3	R_4	$\perp$	R_6	R_5	$\perp$	3

Table 3-1.: All reductions between models.

Partial orders

A partial order can be defined for the minimisation and maximisation of deletions. A graph homomorphism is a mapping between two graphs that respects their structure.

Definition 3.2.1 (Partial order $\prec$) *The partial order $\prec$ is defined on $\mathcal{S}(G, G')$ by $\mu_1 \prec \mu_2$ if an d only if $\forall v \in V, \mu_2(v) \in \{\mu_1(v), \bot\}$ with μ_1 and μ_2 two graph morphisms (i.e $\mu_1 \prec \mu_2$ if some vertices are deleted by μ_2 but merged by μ_1).*

If $\mu_1 \prec \mu_2$ then more vertices are deleted by the morphism μ_2. SEPIs that minimize the number of deleted vertices are minimal for $\prec$. SEPIs that maximize the number of deleted vertices are maximal for $\prec$.

However, SEPIs that minimize the number of vertex deletions do not cover all minimal SEPIs for $\prec$. Conversely, SEPIs that maximize the number of vertex deletions do not cover all maximal SEPIs for $\prec$.

It would also be possible to minimise the number of species deleted while maximising the number of reaction deleted or vice versa.

Definition 3.2.2 (Partial order $\prec_{sr}$) *The partial order $\prec_{sr}$ is defined on $\mathcal{S}(G, G')$ by $\mu_1 \prec_{sr} \mu_2$ if an d only if for all $v \in S(G), \mu_2(v) \in \{\mu_1(v), \bot\}$ and for all $v \in R(G), \mu_1(v) \in \{\mu_2(v), \bot\}$ with μ_1 and μ_2 two graph morphisms (i.e $\mu_1 \prec_{sr} \mu_2$ if some species vertices are deleted by μ_2 while merged by μ_1 or some reaction vertices are deleted by μ_1 while merged by μ_1).*

Only the first partial order will be implemented and tested. If partial order $\prec_{sr}$ is interesting for biologists, its implementation would be straightforward.

Minimisation of the number of deletions

In order to minimise the number of deletions, a MAX-SAT solver is used.

$|S| + |R|$ soft clauses are added representing for each vertex of the starting graph the vertex deletion's negation. All the previous clauses are transformed into hard clauses by adding a weight of $|S| + |R| + 1$ and the solver is used to maximise the number of satisfied clauses.

Maximisation of the number of deletions

Similarly, to maximise the number of deletions, for each vertex of the starting graph a soft clause representing its deletion is added.

No additional variables are needed.

Clauses

I Minimisation of the number of deletions. $\forall u \in S \cup R$, $cl(\neg \mathbf{m}_{u,\bot})$.

Number of clauses: $|S| + |R|$, number of new variables: 0.

II Maximisation of the number of deletions. $\forall u \in S \cup R$, $cl(\mathbf{m}_{u,\bot})$.

Number of clauses: $|S| + |R|$, number of new variables: 0.

Lemma For a MAX-SAT encoding of the minimisation of the number of vertex deletions, with n variables and $m = p + q$ clauses, p corresponding to the number of hard clauses and q to the number of soft clauses, the number of non satisfied clauses r is the minimum number of vertex deletion.

Proof: Each soft clause represents the negation of each vertex deletion. If a soft clause is not satisfied, the corresponding vertex is deleted. A MAX-SAT solver optimises the number of satisfied clauses $m - r$, thus r is the minimum number of vertex deletion. □

3.3. Evaluation

To evaluate the implementation, first it was tested on handwritten models. There structure is well known and results can easily be understood. Then real models of the BioModels database [4] were used.

Handwritten models

Combinations of two Michaelis-Menten

Reduction of Michaelis-Menten reaction, presented as the first example is the most simple and classical example of model reduction. Michaelis-Menten reactions are also present in a lot of complex reactions.

Figure 1-3 shows an expanded form of the reaction, where a complex enzyme-substrate ES is present, as well as a reverse reaction R_{-1} from the complex to initial species substrate and enzyme. Figure 1-4 shows a reduced form of the reaction. Several intermediary forms could be considered, coming examples only take into account intermediary forms with a complex ES and reactions R_1 and R_2 but without reverse reaction R_{-1}.

This first example is important because Michaelis-Menten-like reactions can be found in more complex models with different forms of reduction. Hence, a basic feature any variant of SEPI framework should ensure is to find Michaelis-Menten-like reductions when they exists. Moreover, there is already three possible SEPIs for the most basic example of Michaelis-Menten reduction. If two graphs G and G' contains Michaelis-Menten patterns, the number of SEPIs between those two graphs is exponential in the number of Michaelis-Menten patterns.

Other small examples are combinations of two Michaelis-Menten reactions. For example, with species aS, aE, aP and bS, bE, bP as substrates, enzymes and products of a first motif a and a second motif b. Reactions are combined if some of their species are shared, for example if $aP = bS$ (in this case it will be represented by $aPbS$). Combinations are denoted by shared species, for example $EE - PS$ represents graph of figure 3-4 where two enzymes are shared and product of the first reaction plays a role of substrate of the second. In this graph, both Michaelis-Menten reactions are in reduced form.

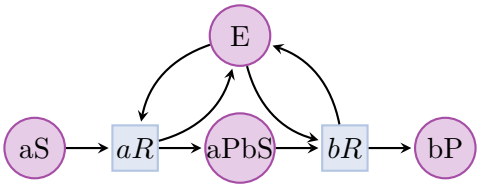

Figure 3-4.: EE-PS: combination of two Michaelis-Menten, with shared species.

There exist 23 combinations of two Michaelis-Menten patterns: 6 combinations with only one shared specie, 12 combinations with two shared species and 5 combinations with three. Some combinations can seem artificial but some of them are found in biological models, for example in Mitogen-Activated Protein Kinases (MAPK) cascade models: PE when the enzyme is the product of a previous reaction, $EE - PS$ for a two-steps phosphorylation with same enzyme, $PS - SP$ for a Michaelis-Menten reaction that is reversible with another enzyme.

Table 3-2 shows the same effects of reducing the number of pairings with Min $\perp$ and Max $\perp$ for combinations of two Michaelis-Menten patterns. It decreases the number of pairs with more than 200 SEPIs and decrease the mean and median of the number of pairings.

	Pairs with SEPIs		Number of SEPIs		No SEPI
Type of pairing	≥ 200	< 200	mean	median	
Normal	99 (2.1%)	508 (10.8%)	20.1	2	4085 (87%)
Min $\perp$	6 (0.3%)	601 (12.8%)	12.4	2	4085 (87%)
Max $\perp$	6 (0.3%)	601 (12.8%)	8.4	2	4085 (87%)

Table 3-2.: Combinations of two Michaelis-Menten.

MAPK cascades

More complex combinations of Michaelis-Menten reactions can be used to model MAPK cascades. Figure 3-5 presents a three-levels model corresponding to reduced form of models described by C.Y. Huang et al. [31] and by A. Levchenko [32].

Graph if figure 3-5 is a combination of 10 Michaelis-Menten reactions. Between expanded and reduced forms, there are more than 2^{10} SEPIs: three SEPIs for each Michaelis-Menten reaction, but when an enzyme is shared it cannot always be deleted and a lot of Michaelis-Menten reactions share an enzyme in MAPK cascades.

Other handwritten models can be considered with one-level or two-level cascades.

	Pairs with SEPIs		Number of SEPIs		No SEPI	Timeouts
Type of pairing	≥ 200	< 200	mean	median		(20 min)
Normal	64	9	25	14	127	10
Min $\perp$	20	54	24	5.5	127	9
Max $\perp$	0	48	18.7	4	127	35

Table 3-3.: Statistics on the sets of pairings for 1, 2 and 3-levels MAPK cascades.

Table 3-3 presents statistics on sets of pairings searched among all one, two and three-levels MAPK cascades. Again, Min $\perp$ and Max $\perp$ filter reductions but Max $\perp$ causes many timeouts.

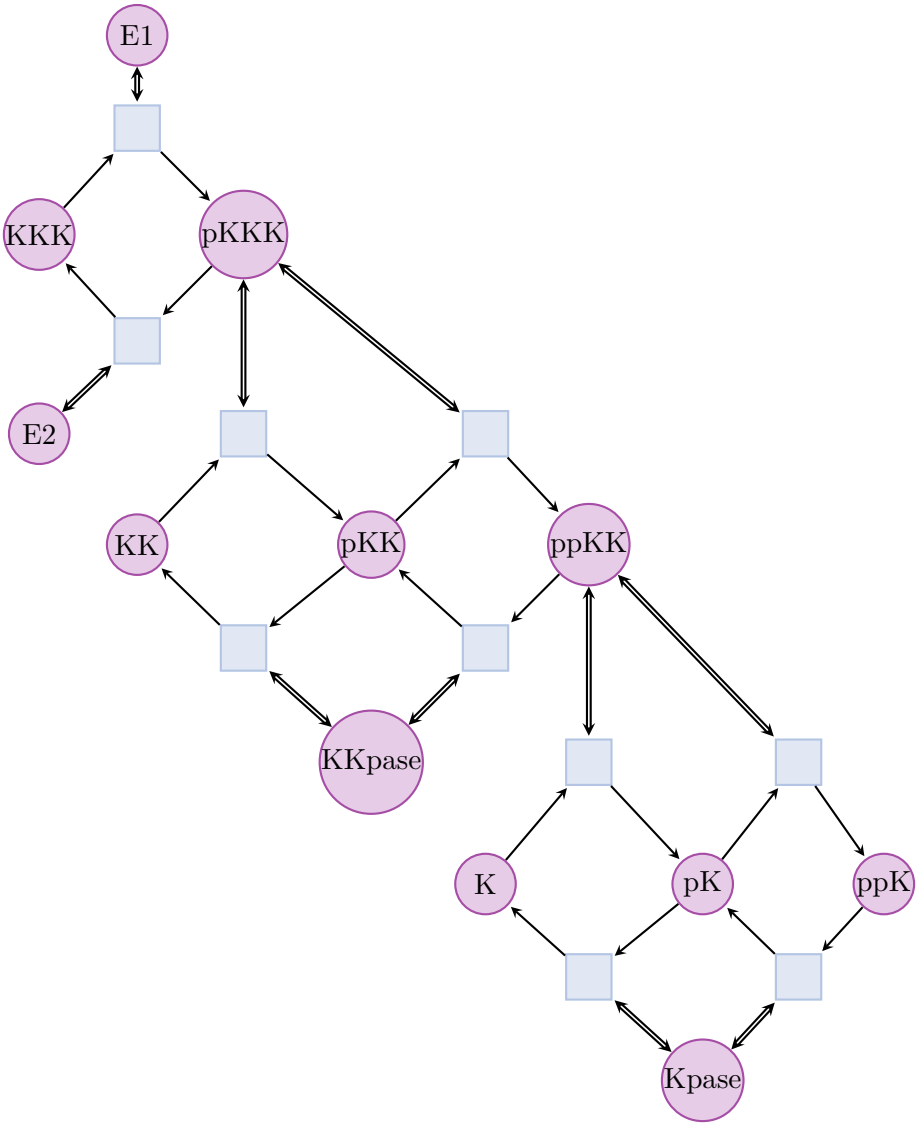

Figure 3-5.: Completely reduced three-levels MAPK cascade.

BioModels

The two cited models from Huang-Ferrel and Levchenko correspond to models 9 and 11 of the publicly-accessible repository BioModels [4]. Previous works on SEPIs [1] and [15] used a set of 42 models from BioModels for their evaluations. These models have been manually clustered by annotations according to the kind of biological process they represent (MAPK cascade, calcium oscillations, circadian clock, cell cycle). Gathered data on reaction graph sizes are presented in table 3-4. Relevancy of SEPI theory was justified because it permits to find the same clustering a priori: more SEPIs are expected intra-class than inter-class.

This set of models is interesting for an evaluation because different kind of observations can be made. SEPI implementation is too relaxed if a lot of reductions are found between models from different classes but is too strict if it reduces the amount of SEPI intra class.

Class	Nb models	Number of vertices			Number of arcs		
		Min	Max	Mean	Min	Max	Mean
Ca Oscillations	11	6	44	14	6	72	21
Cell Cycle	9	20	224	82	31	743	195
Circadian Clock	11	24	68	47	31	93	67
MAPK	11	7	334	62	14	706	124

Table 3-4.: Reaction graph characteristics for each class.

To minimise or maximise the number of vertex deletions, a MAX-SAT solver was used. To select the best one, a benchmark on three different MAX-SAT solvers was made. Those three solvers: RC2 [33], QMaxSat [34] and OpenWBO [35] were selected among many other solvers according to results showed in a MAX-SAT competition [36]. Results of each solver were compared on four classes of BioModels.

Results can be seen in table 3-7. Results are expressed in seconds, timeouts are declared after running the solver for 20 minutes without having results. Additionally table 3-8 presents the time in milliseconds to found first, second and last SEPIs between two given models.

According to these results, the MAX-SAT solver QMaxSat was kept for the implementation in Biocham but after several tests RC2 MAX-SAT solver was better suited for the framework.

After the selection of the MAX-SAT solver, more tests were made on the BioModels database. Results are presented in table 3-5. Additionally, more precise results can be observed in figure A-1 to figure A-12 of the annexes.

On real models, compared to hand-written models, a few observations can be made:

- Globally, the number of timeouts is highly increasing, in particular for maximisation of the number of deletions. Table 3-6 summarises timeouts of each classes. It makes the framework inoperable on big models: there is too few results to interpret them correctly.
- It can still be noted that maximisation of the vertex deletion number filters more reductions.
- Models of different classes behave in distinct ways. Results are better on models representing calcium oscillations than on models depicting circadian clocks.

Drawbacks

However, SEPIs that minimize (respectively maximize) the number of vertex deletion unfortunately do not cover all SEPIs minimal (respectively maximal) for $\prec$. Next figures are an example of SEPIs without biological interpretation that minimises the number of deletion.

Graphs 3-6, 3-7 and 3-8 used in this example could represent a one-step kinase double phosphorylation, with all dephosphorylations. It is composed of four Michaelis-Menten reactions. Figure 3-7 presents the reduced form, Michaelis-Menten reactions are recognizable with double-arrows towards enzymes.

A SEPI μ with minimisation of the number of vertex deletions is expected between both graphs. μ is represented by colours in the figures. Vertices of one colour in figure 3-6 are mapped to the vertex of the same colour in figure 3-7. Colours represent an expected SEPI pairing from expanded form to reduced one, where the four reverse reactions from complexes to enzyme and substrate are deleted, like in Michaelis-Menten reduction.

Class	Type of pairing	SEPIs ≥ 200	SEPIs < 200	SEPIs total	No SEPI	Timeout
Ca (110)	Normal	29	9	38	72	0
	Min ⊥	22	16	38	72	0
	Max ⊥	1	31	32	72	6
Cell (72)	Normal	12	0	12	49	11
	Min ⊥	4	0	4	49	19
	Max ⊥	0	0	0	49	23
Circ (110)	Normal	23	2	25	61	24
	Min ⊥	23	2	25	61	24
	Max ⊥	0	5	5	61	44
MAPK (110)	Normal	39	1	40	60	10
	Min ⊥	26	3	29	60	21
	Max ⊥	0	25	5	60	45
Total (402)	Normal	103	12	115	242	45
	Min ⊥	75	21	96	242	64
	Max ⊥	1	41	42	242	118

Table 3-5.: Number of SEPIs in each class.

	Ca (110)	Cell (72)	Circ (110)	MAPK (110)
normal	0%	15.28%	21.81%	9.09%
min ⊥	0%	26.39%	21.81%	19.09%
max ⊥	5.45%	31.95%	39.99%	38.18%

Table 3-6.: Percentage of timeout in each class.

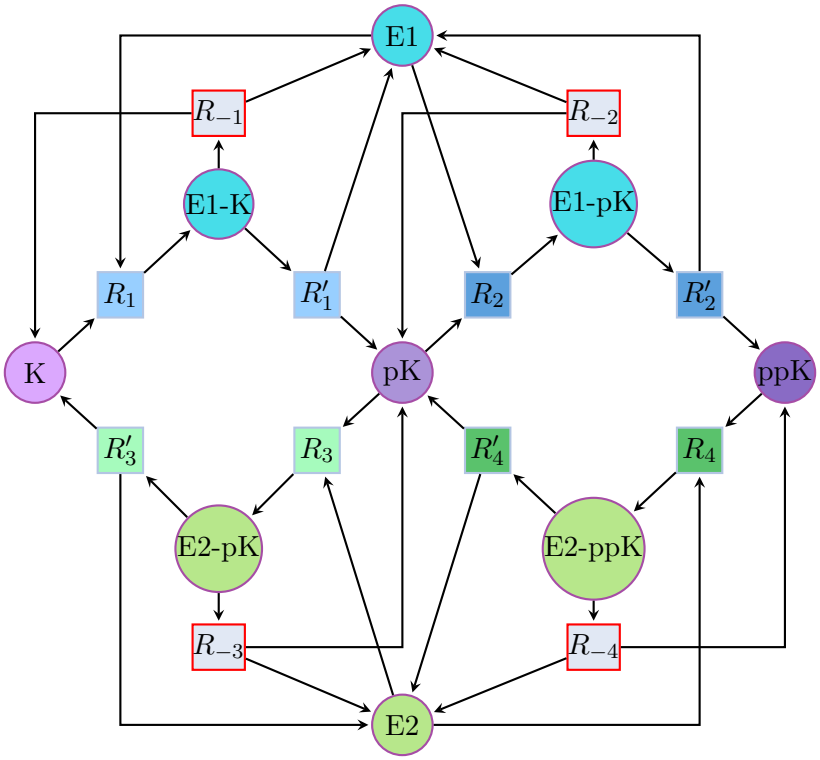

Figure 3-6.: Expected reduction of a one-step kinase phosphorylation.

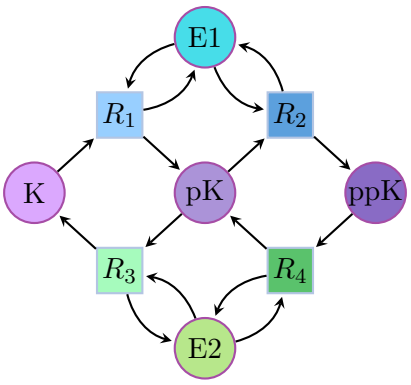

Figure 3-7.: Reduced model of a one-step kinase phosphorylation.

Solver	Calcium Oscillations				Cell Cycle				Circadian Clock				MAPK			
	SEPI	No SEPI	Timeout	Execution time	SEPI	No SEPI	Timeout	Execution time	SEPI	No SEPI	Timeout	Execution time	SEPI	No SEPI	Timeout	Execution time
Bottom minimisation				58 iterations				33 iterations				58 iterations				56 iterations
RC2	38	20	0	11.414	4	9	20	24035.227	28	9	21	25914.057	29	6	21	25268.776
Open-WBO-Gluc	38	20	0	6.120	3	10	20	24503.061	27	9	22	27333.767	29	6	21	25381.987
QMaxSat	38	20	0	6.072	3	10	20	25091.918	25	9	24	29152.416	29	6	21	25568.536
Bottom maximisation				58 iterations				33 iterations				58 iterations				56 iterations
RC2	32	20	6	7215.410	0	9	24	28848.309	5	9	44	53707.753	8	6	42	51847.010
Open-WBO-Gluc	32	20	6	7209.487	0	9	24	28813.223	7	9	42	51941.678	8	6	42	51181.569
QMaxSat	32	21	5	7134.145	0	10	23	28707.418	7	9	42	50820.319	8	6	42	51454.622

Table 3-7.: Solver comparison on Bottom minimisation and maximisation (time in s).

Model	Normal Reductions				Bottom Minimisation									
		Glucose				RC2			OPEN-WBO			QMAXSAT		
	nb SEPI	1st	2nd	last	nb SEPI	1st	2nd	last	1st	2nd	last	1st	2nd	last
043-045	6	13	7	7	2	173	82	(75)	95	14	(11)	61	3	(4)
145-098	200	7	6	196	200	72	69	96	10	7	139	2	2	3
144-008	200	104	97	248	200	691	726	826	577	702	29773	426	447	706
021-171	2	326	808	(1728)	2	383	378	(4191)	603	1871	(2229)	421	787	(3430)
083-084	200	202	87	247	200	806	783	769	608	829	38823	305	303	275
029-027	10	12	12	8	6	103	105	104	19	18	21	3	4	3
011-026	200	4381	4501	61875	200	11745	11690	28507	39617	39614	94653	30641	29546	62250

Table 3-8.: Time to compute the first second and last SEPI for each solver (time in ms).

This pairing is minimal for $\prec$ because if same images for non-deleted vertices are kept, these reverse reactions cannot be merged with other ones. For example, if R_{-1} has to be merged, the only option to preserve an outgoing arc from K is to merge R_{-1} with R'_3, with image R_3. But it is not possible because an arc from R_{-1} to E_1 cannot have an arc from R_3 to E_1 as image.

However, colours in figure 3-8 show a pairing where no vertex is deleted. Hence, the minimal number of vertices is 0, which is not achieved with the previous pairing. This example is problematic because the pairing exhibited by figure 3-8 is very unexpected: entry K and output ppK play a role of enzymes, while all enzymes are merged on pK. This unexpected pairing is also valid with accurate merge restriction.

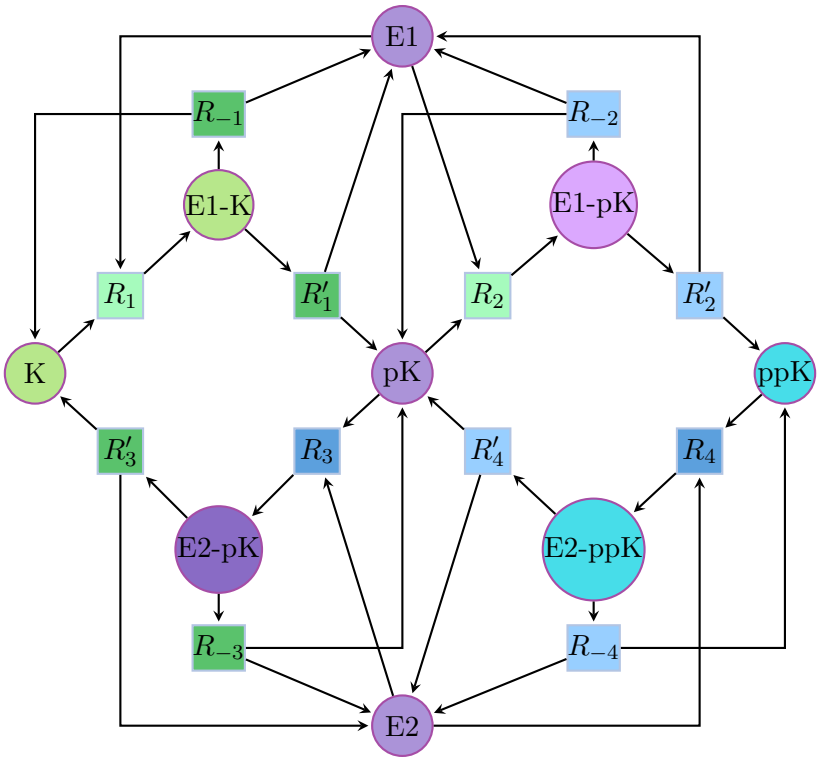

Figure 3-8.: Unexpected reduction that minimizes bottom.

3.4. Conclusion

Adding constraints on the number of vertex deletion gives mixed results. It is reducing a bit the size of the set of SEPIs between two given graphs but the number of timeouts in too big to use this implementation on big models.

After filtration with maximisation or minimisation of the deletion number, SEPIs were more relevant but there were still unwanted SEPIs as the one presented in previous section.

This improvement on its own is not enough but more restrictions on the merge operation can be added. This will be the aim of next sections.

4. Accurate merge restriction

This chapter will present a restriction of the merge operation, which aims to filter SEPIs without biological interpretation between models, in particular, SEPIs that allow to merge vertices far apart in the reaction graph.

In a first section, motivations will be presented. A formal definition of the restriction will be explained in a second section. Then the implementation will be described in a third section and in the last two sections, an evaluation will be performed and conclusions will be drawn.

4.1. Motivations

A major drawback of the SEPI framework is the amount of SEPI found between models. Some existing SEPIs do not have a biological interpretation and the set of SEPIs between two given graphs is often too big to be studied and to be understood according to biologists using the framework. The SEPI framework is not restricting enough.

In the previous chapter extremalisation of the vertex deletion number has been implemented in order to reduce the size of the set of SEPIs found between two given models. But this implementation only reduces the size of a set of SEPIs, it doesn't delete SEPI relations between models.

The SEPI relation is a logical relation defined by the deletion or fusion of vertices. This logical relation between two given graphs doesn't always have a meaning in biology. Therefore, some restrictions to the SEPI relation need to be implemented in order to give more sense. The encoding of previous chapters was still interesting for biologists. It gives insights of structural similarities.

By observing SEPIs drawn between models of opposite classes, it can be seen that sometimes species far apart in the reaction graph are merged together. This is not appropriate because it doesn't have a biological interpretation. It makes more sense to merge species or reactions that are close together. Usually a reduction correspond to the contraction of a chain of reactions or to the reduction of a pattern.

A restriction on the merge operation is thus implemented to allow to merge vertex only when they are close together or when they are close together because their neighbours were also merged together.

Filtering SEPIs is expected to have two main results:

- Reduce the number of SEPIs inter class.

  A high number of SEPIs inter class is not wanted because these SEPIs doesn't always have a biological interpretation. In figure 4-1 it can be seen that there is a lot of SEPIs inter class. And in table 4-1 it can be observed that there is 27.02% of SEPIs inter class compared to 23.66% of SEPIs intra class. Detailed results can be seen in table 4-6 and 4-5.

- Reduce the combinatoric of SEPIs between two given models.

  It is not good to have a high combinatoric of SEPIs between two given models because Biocham is aimed for biologists that study this SEPIs. Thus having more than 200 SEPIs to study is not practicable.

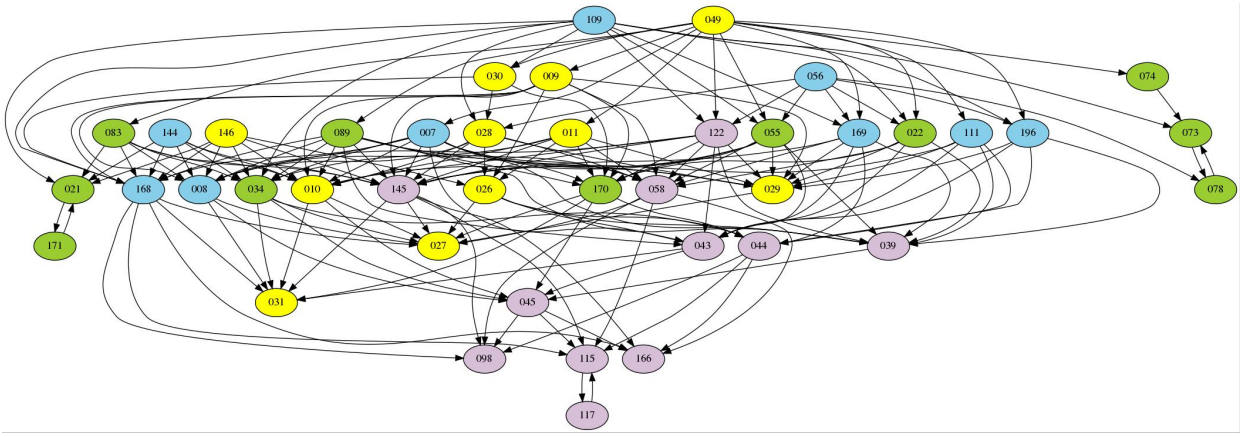

Figure 4-1.: All SEPIs inter and intra classes.

In figure 4-1 each vertex represent a model from BioModels. An edge is drawn from a model to another when there exists a SEPI relation between these two models. Yellow vertices represent models from the class "MAPK", green vertices represent models from the class "Circadian Clock", blue vertices represent models from the class "Cell Cycle" and purple vertices represent models from the class "Calcium Oscillations".

	Pairs	SEPI	No SEPI	Timeouts
Intra class	372	88 23,66%	243 65,32%	41 11,02%
Inter class	1188	321 27.02%	776 65.32%	91 7.66%

Table 4-1.: Number of SEPIs inter and intra class.

Table 4-1 represents the number of SEPIs inter and intra class. The column "Pairs" gives the amount of ordered model pairs for each problem. For example, model 74 to model 73 is considered as one pair, a SEPI relation will be searched from model 74 to model 73 and there exists one. And model 73 to model 74 is considered as another pair. A SEPI relation will be searched from model 73 to model 74 but there is no relation. The column "SEPI" represents the amount of SEPI relations found among all pairs of models. The column "No SEPI" counts the absence of SEPI relations between pairs of models. A timeout is declared after running the SAT solver for 20 minutes, it is represented by the column "Timeout".

4.2. Definitions

A restriction on the merge operation is implemented in this chapter to allow to merge vertices only when they are close to each other or when they are close because their neighbours were also merged together.

For this purpose, the *two-neighbours* relation will be defined and a notion of *good-path* will also be explained in this section and implemented in the following section.

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

To simplify the notation two other sets will be introduced, V and E. $V = S \cup R$ is a set of vertices. $E \subseteq S \times R \cup R \times S$ is a set of un-oriented arcs: $E = \{(u, v) | (u, v) \in A \vee (v, u) \in A\}$.

Let G and G' be two graphs defined by $G = (V, E)$ and $G' = (V', E')$.

Because reaction graphs are bipartite graphs, graphs whose vertices can be divided into two disjoint and independent sets, only vertices of the same type, specie or reaction, can be merged together. A definition of neighbours cannot be used for a merge restriction because neighbours vertices will always be of different types. A *two-neighbours* definition is then introduced to explain this neighbour of neighbour notion.

Definition 4.2.1 (Two-neighbours) *Let a, b be two vertices in V, a and b are two-neighbours if and only if $\exists r \in V$, $((a, r) \in E) \wedge ((b, r) \in E)$.*

$m^*_{u,v}$ will be the notation of the merge operation $m_{u,v}$ where u and v are two-neighbours.

For example in figure 4-2 vertex a and vertex b are *two-neighbours* only because there is a reaction r linking them. Circle vertices represent species and square vertices represent reactions. Edges represent a role in a reaction.

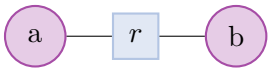

Figure 4-2.: Example of two-neighbours vertices.

But only a *two-neighbours* definition is not enough. It is common to contract a chain of reactions for example, as illustrated in figure 4-3. Vertices A and C are not *two-neighbours* but they participate in reactions $r1$ and $r2$ which have both specie B in common. If reactions $r1$ and $r2$ are merged together, it should be allowed to also merge vertices A and C.

Thus is introduced a definition of *good-path*.

Definition 4.2.2 (Good-path) *Let μ be a SEPI from G to G' and a and b vertices of G such that $\mu(a) = \mu(b)$. There exists a good-path denoted path* C, between a and b if:*

- *$a = b$ then $length(C) = 0$.*
- *there exists $c \in S$, $u, v \in R$ such that:*
  - *$(a, u) \in E$ and $(c, v) \in E$,*
  - *there exists a path* C_1 between u and v,*
  - *there exists a path* C_2 between c and b,*
  - *$\mu(c) = \mu(b)$ and $\mu(u) = \mu(v)$,*

  then $length(C) = lenght(C_1) + length(C_2) + 1$.

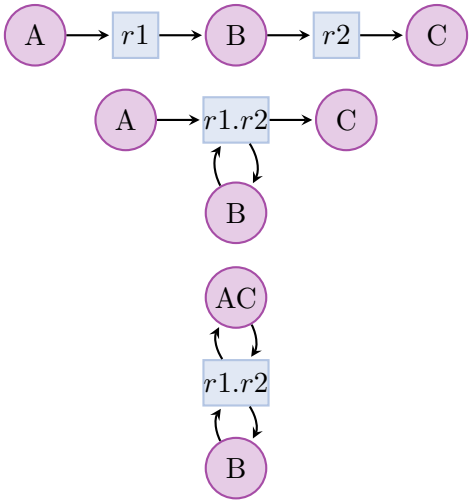

Figure 4-3.: Example of good-path reduction.

Two vertices, for example two specie vertices, are connected by a good-path if there exist two edges, linking them to two reaction vertices, such that those two reactions are also connected by a good-path and they are merged together. This definition is recursive.

Figure 4-4 is an illustration of the *good-path* definition. Dotted edges represent a good-path between two vertices. There exists a good-path between vertices a and b because there exists three vertices u, v and c such that a is related to the reaction u, c is related to the reaction v and there exists a good-path between reactions u and v and there exists a good-path between species c and b.

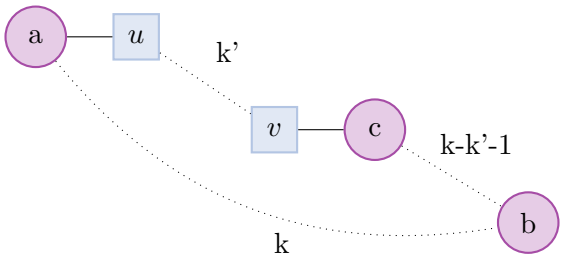

Figure 4-4.: Definition of good-path between vertices a and b.

It is important to associate a length to each path*, defined by the length of smaller intermediary paths* $\mathcal{C}_i$, in order to avoid cyclic situations as described in figure 4-5. It is the reason why a characterization of paths* of length k is needed for the implementation.

Figure 4-5 is an example of cyclic configuration. In this disconnected graph, a graph where there exist two nodes such that no path has those nodes as endpoints, A and B have the same image, and $r1$ and $r2$ as well. The argumentation that there is a path* between A and B with $\mathcal{C}_1$ a path* between $r1$ and $r2$, itself defined thanks to A and B, would be infinitely recursive, no base case is defined. Hence, no length could be defined. It shows why a well-defined length is required for paths*.

A new definition of the SEPI relation can then be defined.

Definition 4.2.3 *A SEPI μ from G to G' is a SEPI* iff for all $a, b \in V$ such that $\mu(a) = \mu(b) \neq \perp$, there exists a path* between a and b.*

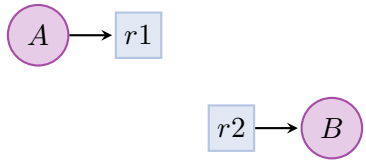

Figure 4-5.: Example of a cyclic configuration.

$G \overset{SEPI^*}{\rightsquigarrow} G'$ is written when there exists a SEPI* from G to G'.

Figure 4-3 was an example of good-path reduction because vertices with the same image through the morphism are linked by a good-path.

Figure 4-6 is an example of invalid good-path reduction. Species A and B can't have the same image in the targeted graph because reactions $r1$ and $r2$ are not merged together.

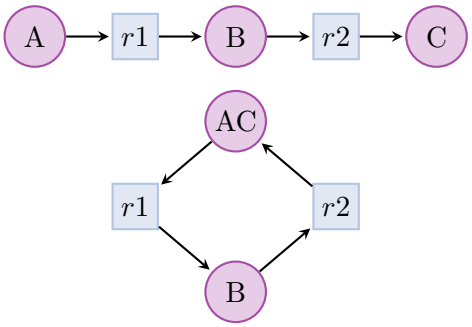

Figure 4-6.: Example of invalid good-path reduction.

Theorem 4.2.1 *Let G and G' be graphs. Then, there exists a SEPI* from G to G' iff there exists a finite sequence of merge* and delete operations that yields a graph isomorphic to G' when applied to G.*

More formally, $G \overset{SEPI^*}{\rightsquigarrow} G' \Leftrightarrow G \rightarrow^*_{m^*d} G'$.

Proof: ($\Rightarrow$) The only if direction will be proved by induction on the number of vertices.

- *base case*: $|V| = |V'|$

No vertex is merged or deleted. A SEPI* gives an isomorphism between G and G', hence the empty sequence works.

- *induction step*: $|V| > |V'|$

Induction hypothesis: the implication is true for any G'' such that $|V''| = |V| - 1$.

- case (i): $\exists c$ such that $\mu(c) = \perp$

G'' will be defined by $G'' := d_c(G)$ and $\widetilde{\mu}$ by:

$$\begin{aligned} \widetilde{\mu} : G'' &\longrightarrow G' \\ x &\longmapsto \mu(x) \end{aligned}$$

$\widetilde{\mu}$ is a SEPI* from G'' to G' and there is a sequence from G'' to G' by induction hypothesis. μ is a SEPI* from G to G' and $G \to_d G'' \to^*_{m^*d} G'$. Hence there is a sequence from G to G'.

- case (ii): $\exists a, b \in V$ such that $\mu(a) = \mu(b) \neq \perp$

Distance $d(a, b)$ will be defined as the length of the smallest path* between a and b in μ.

a and b will be selected such that $d(a, b)$ is the smallest distance in G with associated path* $\{x_0 = a, \ldots, x_n = b\}$. The minimality of $d(a, b)$ implies that $n = 1$ and $length(\mathcal{C}) = 0$, i.e a and b are two-neighbours. G'' is defined by $G'' := m^*_{a,b}(G)$ and $\widetilde{\mu}$ by:

$$\begin{aligned} \widetilde{\mu} : G'' &\longrightarrow G' \\ x &\longmapsto \begin{cases} \mu(a) = \mu(b) & \text{if } x = ab \\ \mu(x) & \text{if not} \end{cases} \end{aligned}$$

$\widetilde{\mu}$ is a SEPI*, with paths* given by μ where vertices a and b are replaced by ab.

$\widetilde{\mu}$ is a SEPI* from G'' to G' and there is a sequence from G'' to G' by induction hypothesis. μ is a SEPI* from G to G' and $G \to_{m*} G'' \to^*_{m^*d} G'$. Hence there is a sequence from G to G'.

($\Leftarrow$) The if direction will be proven by induction on n, the length of the sequence of merge and delete operation between G and G'.

- *base case*: $n = 0$

If the sequence is empty, by definition G and G' are isomorphic, thus an isomorphism μ between them is a SEPI* (since $\mu(a) = \mu(b) \Leftrightarrow a = b$ for any vertices a, b of G).

- *induction step*: $n \neq 0$

Induction hypothesis: the implication is true for any sequence of length smaller or equal to n.

A sequence of length $n + 1$ can be decomposed in a sequence of length n between G and an intermediary graph G'' and a single operation o between G'' and a graph isomorphic to G'. The induction hypothesis gives a SEPI* μ from G to G''.

- case (i): $o = d_u$

With $\mu^{-1}(u)$ the antecedent of u through the morphism μ, $\widetilde{\mu}$ is defined by:

$$\begin{aligned} \widetilde{\mu} : G &\longrightarrow G' \\ x &\longmapsto \begin{cases} \perp & \text{if } x \in \mu^{-1}(u) \\ \mu(x) & \text{if not} \end{cases} \end{aligned}$$

By definition, $\widetilde{\mu}(G)$ is isomorphic to G'. $\forall a, b \in V$, $\mu(a) = \mu(b) \Rightarrow \widetilde{\mu}(a) = \widetilde{\mu}(b)$, hence the same path* works for μ and $\widetilde{\mu}$. Then $\widetilde{\mu}$ is a SEPI*.

- case (ii): $o = m^*_{u,v}$

$\widetilde{\mu}$ is defined by:

$$\widetilde{\mu} : G \longrightarrow G'$$
$$x \longmapsto \begin{cases} uv & \text{if } x \in \{\mu^{-1}(u), \mu^{-1}(v)\} \\ \mu(x) & \text{if not} \end{cases}$$

$\widetilde{\mu}(G)$ is isomorphic to G'. $\forall a, b \in V \backslash \{\mu^{-1}(u), \mu^{-1}(v)\}$, $\mu(a) = \mu(b) \Rightarrow \widetilde{\mu}(a) = \widetilde{\mu}(b)$ hence the same path* works for μ and $\widetilde{\mu}$ for a and b.

For $a, b \in V$ such that $\widetilde{\mu}(a) = \widetilde{\mu}(b) = uv$. If $\mu(a) = \mu(b)$, the same paths* work for μ and $\widetilde{\mu}$ for a and b. u and v are defined by $\widetilde{\mu}(a) = u$ and $\widetilde{\mu}(b) = v$. The possibility to apply merge* operations to u and v implies that there exists $w \in V''$ such that $\{u, w\} \in E''$, and $\{v, w\} \in E''$. Let $\{a', c_1\}$ and $\{b', c_2\}$ be antecedents of edges $\{u, w\}$ and $\{v, w\}$, with $\mu(a') = u$, $\mu(b') = v$ and $\mu(c_1) = \mu(c_2) = w$. The concatenation of paths* between a and a', c_1 and c_2, and b and b' given by μ gives a path* from a to b for $\widetilde{\mu}$. Then $\widetilde{\mu}$ is a SEPI*.

This conclude the proof that $G \overset{SEPI^*}{\rightsquigarrow} G' \Leftrightarrow G \rightarrow^*_{m^*d} G'$.

□

Property 4.2.1 *Merge* and delete operations enjoy the commutation and association properties of figure 4-7, where the solid arrow represents universal quantification and the dashed arrow represents existential quantification. Merge* and delete operation are commutative because changing the order of the operands does not change the result.*

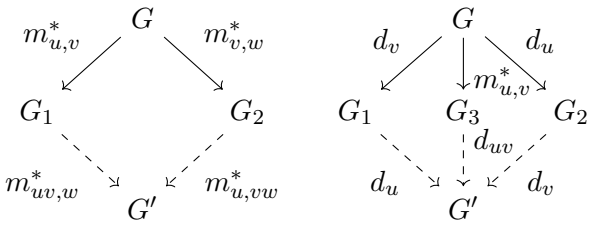

Figure 4-7.: Properties of merge* and delete operations.

Observation: SEPI* is not a well-quasi-order.

Proof: SEPI* relation is more restrictive than SEPI. Hence, the same example of infinite antichain given for SEPI by S. Gay et al. [16] works. □

Lemma: The SEPI*-decision problem is NP-complete.

This lemma was proved by Eva Philippe.

Proof: The proof of NP-completeness found by C. Solnon [1] only needs a very slight modification to apply to SEPI*: adding a vertex w that makes the merged vertices be two-neighbours.

The idea is to reduce from exactly-k-set-covering problem [28], which is NP-complete, to SEPI*. This exactly-k-set-covering problem is defined by:

Instance. A set E, subsets $U \subseteq \mathcal{P}(E)$, and an integer $k \leq |U|$.

Question. Is there a family $U* \subseteq U$ such that $U* = k$ and $\bigcup_{s \in U*} s = E$?

(E, U, k) is an instance of exactly-k-set-covering, $E = \{e_1, \ldots, e_n\}$ and $U = \{u_1 \ldots, u_m\}$.

$G = (S, R, A)$ is defined with $S = \{u_1, \ldots, u_m\} \cup \{e_1, \ldots, e_n\} \cup \{w\}$, $R = \{r_1, \ldots, r_m\}$, and $A = \{(u_i, r_i) \mid 1 \leq i \leq m\} \cup \{(r_i, e_j) \mid e_j \in u_i\} \cup \{(r_i, w) \mid 1 \leq i \leq m\} \cup \{(w, r_i) \mid 1 \leq i \leq m\}$.

$G' = (S', R', A')$ is defined with $S' = \{s'_1, \ldots, s'_m\} \cup \{e_1, \ldots, e_n\}, R' = \{r'\}$, and $A' = \{(s'_i, r') \mid 1 \leq i \leq k\} \cup \{(r', e_j) \mid 1 \leq j \leq n\}$.

It will be proved that the exactly-k-set-covering problem has a positive answer if and only if there is a reaction graph SEPI* from G to G'.

$\Rightarrow$ Hypothesis: $U^* = \{u^*_1, \ldots, u^*_k\}$ is a solution of (E, U, k).

If $U \setminus U^* = \{u^*_{k+1}, \ldots, u^*_m\}$. Then μ defined by $u_j = u^*_i \mapsto s'_i$, $r_i \mapsto r'$, $e_i \mapsto e_i$ is a reaction graph SEPI* from G to G' (merged vertices all belong to R and have w as a common neighbour).

$\Leftarrow$ Hypothesis: there is a reaction graph SEPI* μ from G to G'.

With $\mu^{-1}(r') = \{r_{i_1}, \ldots, r_{i_k}\}$ and $U^* = \{u_{i_1}, \ldots, u_{i_k}\}$, U^* is a solution of the covering problem. First, U^* covers E. Indeed, let $e \in E$. Then the arc $(r', \mu(e))$ is covered in G' , by some arc (r_i, e_1) in G. $r_i \in \mu^{-1}(r')$, but is e_1 always the vertex e ? Yes: since there are the same number of s-vertices with no exiting arcs in G and G', μ induces a bijection between s-vertices with no exiting arcs of G and G', so $e_1 = e$. Which proves that e is covered by the subset corresponding to r_i . Next, U^* has k elements. Vertex types force arcs (u_i, r_i) to be the only ones that can cover the (s'_i, r'). So the preimages of the s'_i are some u_i , and there must be exactly k such u_i because of the bijection on s-vertices with no entering arcs. U^* is exactly those u_i . The coding being polynomial, this concludes the proof of reduction from exactly-k-set-covering to the reaction graph SEPI* decision problem, and the proof of NP-completeness. □

Figure 4-8 is, for example, an instance of the set covering problem (E, U, k) such that $E = \{a, b, c, d, e, f\}$ and $U = \{\{a, c, d\}, \{a, b, d\}, \{c, f\}, \{e, f\}\}$. Figure 4-8(a) displays the corresponding source graph, figure 4-8(b) the target graph corresponding to $k = 3$, and figure 4-8(c) the target graph corresponding to $k = 2$.

The graph 4-8(a) may be transformed into graph 4-8(b) by deleting the reaction vertex associated with $\{a, c, d\}$ and by merging the three other reaction vertices, respectively associated with $\{a, b, d\}$, $\{c, f\}$ and $\{e, f\}$. These vertices are two-neighbours thanks to vertex w, which is deleted. This corresponds to the solution of the set covering problem such that the three selected subsets are $\{a, b, d\}$, $\{c, f\}$ and $\{e, f\}$.

However, graph 4-8(a) cannot be transformed into graph 4-8(c), as the set covering problem instance has no solution for $k = 2$.

4.3. Implementation

Despite the NP-completeness, the SEPI* problem can be encoded in SAT instances like SEPI. To implement a restriction of the merge operation, constraints are added to the SEPI framework. This new framework is called the SEPI* framework.

For this implementation, new variables need to be introduced and additional clauses need to be written.

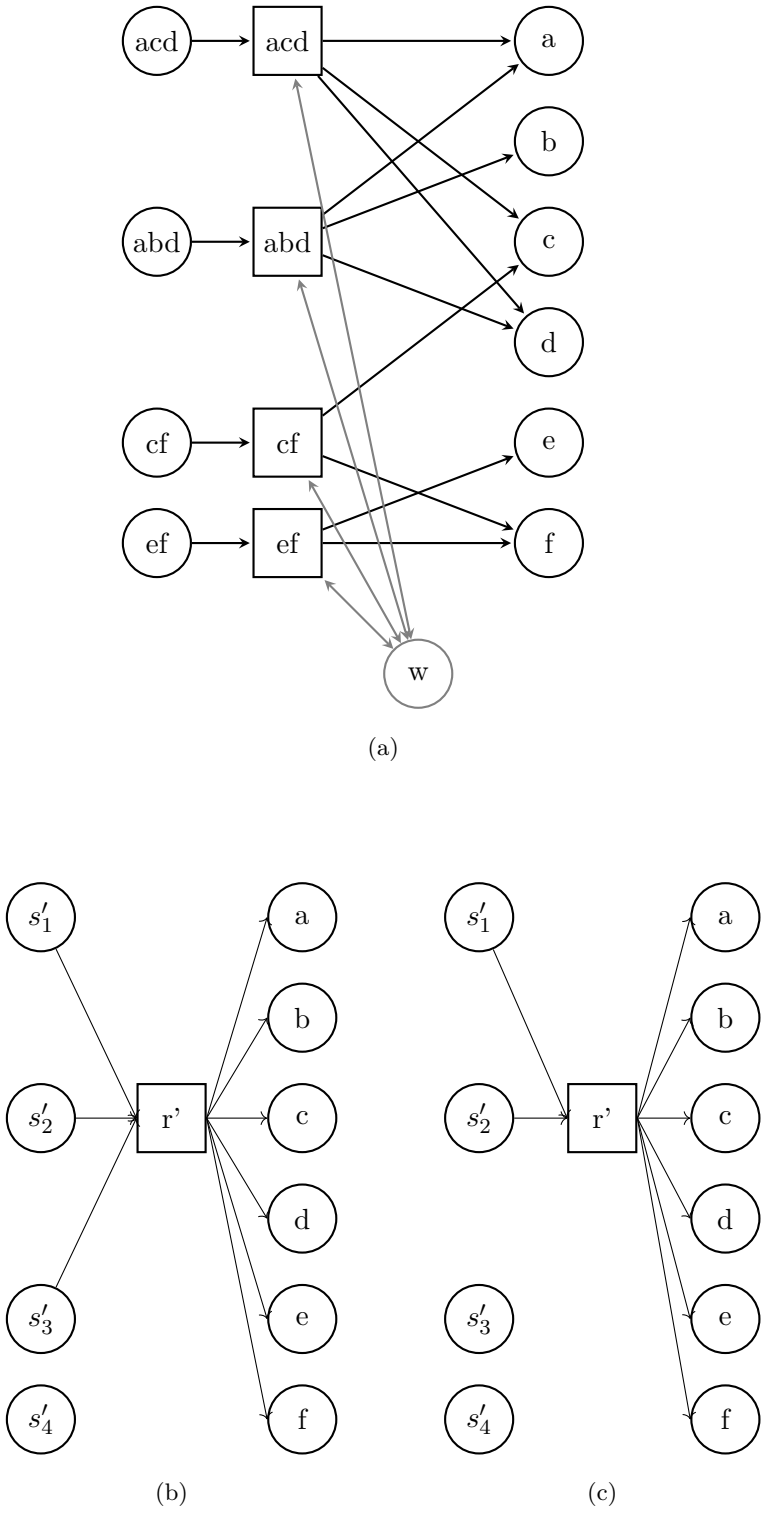

Figure 4-8.: Instance of the set covering problem (E, U, k).

New variables

The same notations as the one of section 2.5.1 are used in this section. As a reminder, m is a graph morphism from G to G', $\mathbf{m}_{a,y} = 1$ if and only if $m(a) = y$.

The following variables will be introduced, in addition to the variables of the SEPI framework.

For all $a, b \in V$, $\forall k \in [\![0, min(|S|, |R|)]\!]$, $\mathbf{p}_{a,b,k}$ represents the existence of a good-path of length k between a and b. $\mathbf{p}_{a,b,k} = 1$ if and only if there exists a good-path of length k between a and b.

$\mathbf{m}_{a,b,y} = 1$ if and only if $m(a) = y$ and $m(b) = y$. Vertices a and b have the same image y through the morphism m.

$\mathbf{p}_{a,b,k_1,c,d,k_2} = 1$ if and only if $\mathbf{p}_{a,b,k_1} = 1$ and $\mathbf{p}_{c,d,k_2} = 1$.

Variables $\mathbf{m}_{a,b,y}$ and $\mathbf{p}_{a,b,k_1,c,d,k_2}$ are introduced mainly to remove conjunctions from disjunctions of literals.

Clauses

The following clauses will be introduced, in addition to the clauses of the SEPI framework.

Inductive merge definition clauses define the new variable $\mathbf{m}_{a,b,y}$. This variable will be required to remove the conjunction from the clause *Good-path of length k*.

The definition of a good-path will be made recursively. *Good-path of length zero* clauses define the base case.

To define the inductive step the variable $\mathbf{p}_{a,b,k_1,c,d,k_2}$ need to be introduced to remove a conjunction of literals. Clauses *inductive good-path definition* define this new variable.

Good-path of length k clauses are the recursive definition's inductive step.

*Path** clauses are linking together the merge definition and the good-path definition, restricting the merge definition.

I Inductive merge definition:

i $F_{merge-i} := \bigwedge_{\substack{\forall a,b \in V \\ \forall y \in V' \cup \{\bot\}}} cl(\neg \mathbf{m}_{a,b,y} \vee \mathbf{m}_{a,y})$,

ii $F_{merge-ii} := \bigwedge_{\substack{\forall a,b \in V \\ \forall y \in V' \cup \{\bot\}}} cl(\neg \mathbf{m}_{a,b,y} \vee \mathbf{m}_{b,y})$,

iii $F_{merge-iii} := \bigwedge_{\substack{\forall a,b \in V \\ \forall y \in V' \cup \{\bot\}}} cl(\neg \mathbf{m}_{a,y} \vee \neg \mathbf{m}_{b,y} \vee \mathbf{m}_{a,b,y})$.

II Good-path of length zero:

i $F_{path-0-i} := \bigwedge_{\forall a,b \in V} cl(\mathbf{p}_{a,a,0})$,

ii $F_{path-0-ii} := \bigwedge_{\forall a,b \in V} cl(\neg \mathbf{p}_{a,b,0})$.

III Inductive good-path definition:

i $F_{inductive-path-i} := \bigwedge_{\substack{\forall a,b,c,d \in V \\ \forall k_1,k_2 \in [\![0,(min(|S|,|R|)-1)]\!]}} cl(\neg \mathbf{p}_{a,b,k_1,c,d,k_2} \vee \mathbf{p}_{a,b,k_1}),$

ii $F_{inductive-path-ii} := \bigwedge_{\substack{\forall a,b,c,d \in V \\ \forall k_1,k_2 \in [\![0,(min(|S|,|R|)-1)]\!]}} cl(\neg \mathbf{p}_{a,b,k_1,c,d,k_2} \vee \mathbf{p}_{c,d,k_2}),$

iii $F_{inductive-path-iii} := \bigwedge_{\substack{\forall a,b,c,d \in V \\ \forall k_1,k_2 \in [\![0,(min(|S|,|R|)-1)]\!]}} cl(\neg \mathbf{p}_{a,b,k_1} \vee \neg \mathbf{p}_{c,d,k_2} \vee \mathbf{p}_{a,b,k_1,c,d,k_2}).$

IV Good-path of length k:

i $F_{path-k-i} := \bigwedge_{\substack{\forall a,b \in V \\ \forall k \in [\![1,min(|S|,|R|)]\!]}} cl(\neg \mathbf{p}_{a,b,k} \vee (\bigvee_{y \in V' \cup \{\perp\}} \mathbf{m}_{a,b,y})),$

ii $F_{path-k-ii} := \bigwedge_{\substack{\forall a,b \in V \\ \forall k \in [\![1,min(|S|,|R|)]\!]}} cl(\neg \mathbf{p}_{a,b,k} \vee \bigvee_{\substack{c \in V \\ c \neq a}} \bigvee_{\substack{u \in V \\ (a,u) \in E}} \bigvee_{\substack{v \in V \\ (c,v) \in E}} \bigvee_{\substack{k' \in \mathbb{N} \\ k' < k}} \mathbf{p}_{u,v,k',c,b,k-k'-1}).$

V Path*:

$F_{path*} := \bigwedge_{\substack{\forall a,b \in V \\ \forall y \in V' \cup \{\perp\}}} cl(\neg \mathbf{m}_{a,b,y} \vee (\bigvee_{k \in [\![0,min(|S|,|R|)]\!]} \mathbf{p}_{a,b,k})).$

The formula is then defined as $F_{SEPI*} := F_{SEPI} \wedge F_{merge-i} \wedge F_{merge-ii} \wedge F_{merge-iii} \wedge F_{path-0-i} \wedge F_{path-0-ii} \wedge F_{inductive-path-i} \wedge F_{inductive-path-ii} \wedge F_{inductive-path-iii} \wedge F_{path-k-i} \wedge F_{path-k-ii} \wedge F_{path*}$.

Optimisations

To introduce fewer variables and to write fewer clauses, some optimisations can be made. An order on graph G vertices is introduced.

I Inductive merge definition.

Clauses are only written when $a < b$ and $type(a) = type(b) = type(y)$.

II Good-path of length zero.

Clauses are only written when $a \leq b$ and $type(a) = type(b)$.

III Inductive good-path definition.

Clauses are only written when $a \leq b, c \leq d$, $type(a) = type(b), type(c) = type(d)$ and $type(a) \neq type(c)$.

IV Good-path of length k.

Clauses are only written when $a < b$ and $type(a) = type(b)$.

V Path*.

Clauses are only written when $a < b$ and $type(a) = type(b) = type(y)$.

Number of variables

Optimisation reduced the number of variables.

For $\mathbf{m}_{a,b,y}$, $\frac{|S|\times(|S|-1)}{2} \times (|S'|+1) + \frac{|R|\times(|R|-1)}{2} \times (|R'|+1)$ variables are introduced.

For $\mathbf{p}_{a,b,k}$, $(\frac{|S|\times(|S|-1)}{2} + |S| + \frac{|R|\times(|R|-1)}{2} + |R|) \times (min(|S|,|R|)+1)$ variables are introduced.

For $\mathbf{p}_{a,b,k_1,c,d,k_2}$, $(\frac{|S|\times(|S|-1)}{2} + |S|) \times (\frac{|R|\times(|R|-1)}{2} + |R|) \times 2 \times (min(|S|,|R|)+1)^2$ variables are introduced.

Number of clauses

The optimisation also reduced the number of clauses added for the definition of SEPI*:

I Inductive merge definition: $3 \times (\frac{|S|\times(|S|-1)}{2} \times (|S'|+1) + \frac{|R|\times(|R|-1)}{2} \times (|R'|+1))$ clauses.

II Good-path of length zero: $\frac{|S|\times(|S|-1)}{2} + \frac{|R|\times(|R|-1)}{2} + |V|$ clauses.

III Inductive good-path definition:
$3 \times (\frac{|S|\times(|S|-1)}{2} + |S|) \times (\frac{|R|\times(|R|-1)}{2} + |R|) \times 2 \times (min(|S|,|R|)+1)^2$ clauses.

IV Good-path of length k: $2 \times (\frac{|S|\times(|S|-1)}{2} + \frac{|R|\times(|R|-1)}{2}) \times min(|S|,|R|)$ clauses.

V Path*: $\frac{|S|\times(|S|-1)}{2} \times (|S'|+1) + \frac{|R|\times(|R|-1)}{2} \times (|R'|+1)$ clauses.

Even with the optimisation, the number of clauses to high to compute the SEPI* on big models. To reduce the number of clauses, the length of the good-path can be restricted to a fixed number. It has been implemented by replacing $min(|S|,|R|)$ by a smaller number in the clauses. $min(|S|,|R|)$ is the estimation of the maximal length of the good-path.

4.4. Evaluation

This section discusses an evaluation of this new framework and problems it raised.

The problem of reproducibility

Biocham version and subgraph epimorphism implementation changed since the last results of Steven Gay's thesis [1]. The thesis had good results with the number of SEPI connections inter classes but it was not reproducible with the new implementation. Even by using the same old models from 2015 the best current result is an inter class connection of 29% instead of the 9% expected [1].

Tests were performed on the same models, the ones from 2015, with the same timeout of 20 minutes and using the same SAT solver Glucose. Table 4-2 compares intra class results. Table 4-3 shows inter class results from 2019. Precise results from 2015 on inter class performance are not available, the number of 9% for the number of SEPIs inter class is the only indication.

Table 4-2 and table 4-3 present results on old models. These models are also ordered in four classes, *Ca* represents the class Calcium Oscillations, *Cell* represents the class Cell Cycle, *Circ* represents the class Circadian Clock and *MAPK* represents the class Mitogen-Activated Protein

		2015 results			2019 results		
		SEPI	No SEPI	Timeouts	SEPI	No SEPI	Timeouts
Ca	(110)	38	72	0	38	72	0
		34.55%	65.45%	0%	34.55%	65.45%	0%
Cell	(72)	12	51	9	12	49	11
		16.67%	70.83%	12.50%	16.67%	68.06%	15.27%
Circ	(110)	37	73	0	25	61	24
		33.64%	66.36%	0%	22.73%	55.45%	21.82%
MAPK	(110)	38	63	9	40	60	10
		34.55%	57.27%	8.18%	36.36%	54.55%	9.09%
Total	(402)	125	259	18	115	242	45
		31.09%	64.43%	4.48%	28.61%	60.20%	11.19%

Table 4-2.: Comparison with results of 2015: number of SEPI relations intra class.

			2019 results		
			SEPI	No SEPI	Timeouts
Ca	Cell	(198)	79	114	5
			39.90%	57.57%	2.53%
Ca	Circ	(242)	100	133	9
			41.32%	54.96%	3.72%
Ca	MAPK	(242)	82	155	5
			33.88%	64.05%	2.07%
Cell	Circ	(198)	31	129	38
			15.66%	65.15%	19.19%
Cell	MAPK	(198)	48	115	35
			24.24%	58.08%	17.68%
Circ	MAPK	(242)	55	147	40
			22.73%	60.74%	16.53%
Total		(1320)	395	793	132
			29.92%	60.08%	10.00%

Table 4-3.: Comparison with results of 2015: number of SEPI relations inter class.

Kinases. The number in brackets represents the number of model pairs in the class (or couple of classes). Again, a timeout if declared when the SAT solver didn't give a result after 20 minutes.

It can be observed in table 4-2 that the previous amount of timeouts was much lower. The number of SEPI relations between pairs of models was also higher. It is a bit problematic to not find the same results in 2019.

In table 4-3 a really high number of SEPIs inter class can be noted. It is more than three times more compared to previous results.

Benchmark for further evaluation

Because it is difficult to reproduce results from 2015, benchmarks were computed again on new versions of models from BioModels. These models were curated with the latest version of Biocham and new tests are performed with the latest version of the SEPI framework too.

Table 4-4 presents characteristics of the recent version of models. It can be noticed that the number of edges is higher compared to old models (same characteristics on previous models were displayed in table 3-4). And two models of the Cell Cycle class are missing, they were too big to be curated by the latest version of Biocham.

		Number of vertices			Number of arcs		
Class	Nb models	Min	Max	Mean	Min	Max	Mean
Ca Oscillations	11	8	44	16	11	72	28
Cell Cycle	7	20	189	70	38	364	155
Circadian Clock	11	25	82	55	38	130	86
MAPK	11	10	334	63	29	744	137

Table 4-4.: Reaction graph characteristics for each class.

Results presented in table 4-5 and table 4-6 will be used as benchmarks for further tests. The important result of table 4-5 is the number of timeouts (11.02%) which need to be decreased. Important results of table 4-6 is the number of SEPIs inter class (27.02%) and the number of timeouts (7.66%). Both also need to be decreased.

To be able to observe a diminution of the number of SEPIs between two given models, it is also specified in table 4-5 the number of SEPI sets that are below a size of 200. Results on the new set of models are similar to results on previous models with the same SEPI implementation. With just a lower amount of timeouts and SEPI relations inter class and intra class.

	SEPI			No SEPI	Timeout
Class	≥ 200	< 200	total		
Ca (110)	20	6	26 (23.64%)	83 (75.45%)	1 (00.91%)
Cell (42)	8	0	8 (19.05%)	28 (66.67%)	6 (14.28%)
Circ (110)	16	3	19 (17.27%)	70 (63.64%)	21 (19.09%)
MAPK (110)	35	0	35 (31,82%)	62 (56,36%)	13 (11,82%)
Total (372)	79	9	88 (23,66%)	243 (65,32%)	41 (11,02%)

Table 4-5.: SEPIs intra class without restriction.

Results inter class			SEPIs	No SEPIs	Timeouts
Ca	Cell	(154)	43 (27.92%)	109 (70.78%)	2 (01.30%)
Ca	Circ	(242)	100 (41.32%)	136 (56.20%)	6 (02.48%)
Ca	MAPK	(242)	74 (30.58%)	162 (66.94%)	6 (02.48%)
Cell	Circ	(154)	22 (14.29%)	109 (70.78%)	23 (14.93%)
Cell	MAPK	(154)	30 (19.48%)	105 (68.18%)	19 (12.34%)
Circ	MAPK	(242)	52 (21.49%)	155 (64.05%)	35 (14.46%)
Total		(1188)	321 (27.02%)	776 (65.32%)	91 (7.66%)

Table 4-6.: SEPIs inter class without restriction.

Accurate merge restriction evaluation

The implementation of this new framework needs to define $\mathcal{O}(n^6)$ variables $\mathbf{p}_{a,b,k_1,c,d,k_2}$ (with $n = max(|V|, |V'|)$). All this variables are required to replace $\mathbf{p}_{a,b,k_1} \wedge \mathbf{p}_{c,d,k_2}$ in the clause characterizing path* of length k. Therefore, for big models even writing the clauses before trying to resolve the SAT problem takes too much time.

This implementation is not usable in practice.

4.5. Conclusion

The definition of good-path is honourable in a logical point of view. It could achieve the goal of filtering unwanted SEPIs and it brings pleasant properties. But the implementation is very heavy. This implementation was the most optimum found and even with some optimisations and by reducing the maximal length of the good-path between each vertex it is not usable in practice.

Another implementation needs to be found. A lighter one that could be used on big models of the BioModels database. This will be the goal of the next chapter.

5. Strict two-neighbours restriction

This chapter presents a second merge restriction implemented, called *strict two-neighbours merge restriction*. A first part explains why a new implementation is necessary, a second part in this chapter presents this implementation. Finally, this chapter ends with an evaluation of this new implementation and a conclusion.

5.1. Motivations

Previous implementation was too complex and too slow to be computed on big models of BioModels. For example, the biggest model of MAPK has 334 vertices and 744 edges.

Table 5-1 and table 5-2 summarise the complexity of the previous implementation. This implementation is based on the SEPI framework, new clauses and new variables are added to already existing clauses and variables of the SEPI framework to restrict the merge operation.

As a reminder, |S| (respectively |S'|) is the number of species of the initial graph (respectively targeted graph). |R| (respectively |R'|) is the number of reactions of the initial graph (respectively targeted graph). |V| is the number of vertices, $|V| = |S| + |R|$ and $|V'| = |S'| + |R'|$. Orders of magnitude are given with $n = \max(|V|, |V'|)$.

New variables	Exact number
$\mathbf{m}_{a,b,y}$	$\frac{\|S\|\times(\|S\|-1)}{2} \times (\|S'\|+1) + \frac{\|R\|\times(\|R\|-1)}{2} \times (\|R'\|+1)$
$\mathbf{p}_{a,b,k}$	$(\frac{\|S\|\times(\|S\|-1)}{2} + \|S\| + \frac{\|R\|\times(\|R\|-1)}{2} + \|R\|) \times (\min(\|S\|, \|R\|)+1)$
$\mathbf{p}_{a,b,k_1,c,d,k_2}$	$(\frac{\|S\|\times(\|S\|-1)}{2} + \|S\|) \times (\frac{\|R\|\times(\|R\|-1)}{2} + \|R\|) \times 2 \times (\min(\|S\|, \|R\|)+1)^2$
	Order of magnitude
$\mathbf{m}_{a,b,y}$	$\mathcal{O}(n^3)$
$\mathbf{p}_{a,b,k}$	$\mathcal{O}(n^3)$
$\mathbf{p}_{a,b,k_1,c,d,k_2}$	$\mathcal{O}(n^6)$

Table 5-1.: Number of variables added by the accurate merge restriction.

This implementation needs to define $\mathcal{O}(n^6)$ variables $\mathbf{p}_{a,b,k_1,c,d,k_2}$, which are required to replace $\mathbf{p}_{a,b,k_1} \wedge \mathbf{p}_{c,d,k_2}$ in the clause characterizing *path* of length k*. Hence, for big models even writing clauses requires an unreasonable amount of time. This implementation is not usable in practice for big models.

Unfortunately, a better optimised implementation for the accurate good-path merge restriction was not found. It doesn't seem to be possible to find local properties implied by the existence of a good-path between two vertices: there can be arbitrary size-neighbourhoods of vertices with the same image that have nothing in common.

New clauses	Exact number
Inductive merge def.	$3 \times (\frac{\lvert S\rvert \times (\lvert S\rvert -1)}{2} \times (\lvert S'\rvert + 1) + \frac{\lvert R\rvert \times (\lvert R\rvert -1)}{2} \times (\lvert R'\rvert + 1))$
Path* of length zero	$\frac{\lvert S\rvert \times (\lvert S\rvert -1)}{2} + \frac{\lvert R\rvert \times (\lvert R\rvert -1)}{2} + \lvert V\rvert$
Inductive path* def.	$3 \times (\frac{\lvert S\rvert \times (\lvert S\rvert -1)}{2} + \lvert S\rvert) \times (\frac{\lvert R\rvert \times (\lvert R\rvert -1)}{2} + \lvert R\rvert) \times 2 \times (\min(\lvert S\rvert, \lvert R\rvert) + 1)^2$
Path* of length k	$2 \times (\frac{\lvert S\rvert \times (\lvert S\rvert -1)}{2} + \frac{\lvert R\rvert \times (\lvert R\rvert -1)}{2}) \times min(\lvert S\rvert, \lvert R\rvert)$
Path*	$\frac{\lvert S\rvert \times (\lvert S\rvert -1)}{2} \times (\lvert S'\rvert + 1) + \frac{\lvert R\rvert \times (\lvert R\rvert -1)}{2} \times (\lvert R'\rvert + 1)$

	Order of magnitude
Inductive merge def.	$\mathcal{O}(n^3)$
Path* of length zero	$\mathcal{O}(n^2)$
Inductive path* def.	$\mathcal{O}(n^6)$
Path* of length k	$\mathcal{O}(n^3)$
Path*	$\mathcal{O}(n^3)$

Table 5-2.: Number of clauses added by the accurate merge restriction.

Figure 5-1 is an example of this impossibility. It would make sense to merge together reactions r_1 and r_2 and merge together reactions r_3 and r_4. And then merge together A, C and E. These species are far apart from each other and species B and D don't have the same image in the targeted graph. Thus, merging A, C and E can't be defined locally.

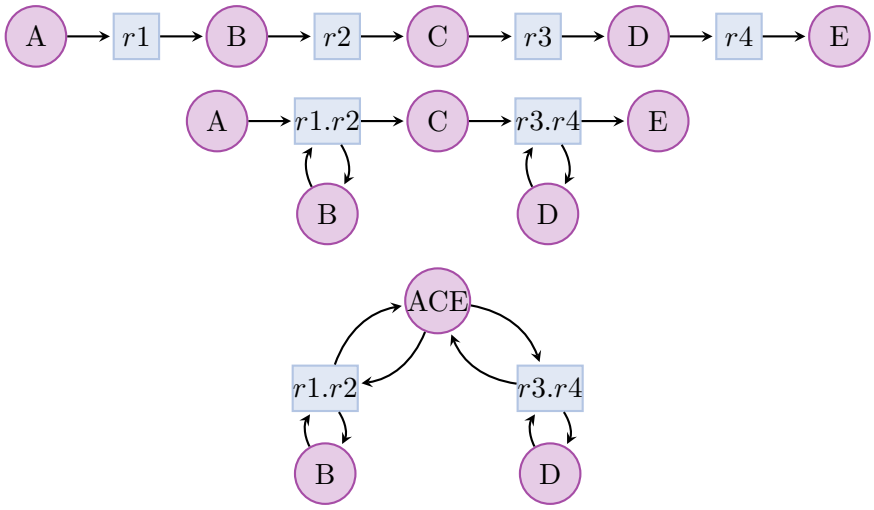

Figure 5-1.: Counter example for local definition.

A local definition is regardless implemented. The good-path merge restriction cannot be used on models of BioModels anyway. This implementation is more restrictive and removes SEPIs that had a biological interpretation but was still considered as a good compromise. Strict two-neighbours implementation is equivalent to $maxpath = 1$ in the previous implementation.

Figure 5-2 is an example of simple not allowed vertex merge that would have been nice to keep

but will be filtered out by the strict two-neighbours merge implementation.

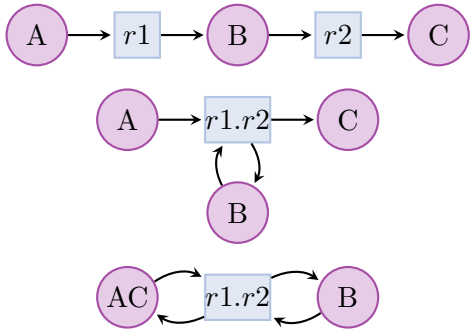

Figure 5-2.: Example of not allowed reduction.

Another major drawback of this stronger restriction is the loss of SEPI's transitivity property.

5.2. Implementation

Strict two-neighbours merge restriction is a much stronger restriction. With the previous definition of good-path it corresponds to a maximal length of 1. It is a local constraint, two vertices of the initial graph can have the same image in the targeted graph if and only if they are two-neighbours in the initial graph. This restriction doesn't need recursive nor dynamic definitions.

Michaelis-Menten reduction is compatible with this restriction.

Clauses

These clauses are added to the already existing SEPI implementation's clauses. With this stronger restriction, no new variables need to be defined and it is introducing less than $|V|^2 \times |V'|$ new clauses. $neigh(a, b)$ is written when a and b are two-neighbours.

Clauses are defined by $F_{merge} := \bigwedge_{\substack{\forall a,b \in V \neg neigh(a,b) \\ \forall y \in V'}} cl(\neg \mathbf{m}_{a,y} \vee \neg \mathbf{m}_{b,y})$.

Then the formula is then defined as $F_{SEPI-strict} := F_{SEPI} \wedge F_{merge}$.

Pseudocode

Before writing clauses of subsection above, the relation *two-neighbours* between vertices need to be identified.

As a reminder, the *two-neighbours* definition is the following: a and b are *two-neighbours* if and only if $\exists r \in V$ such that $((a, r) \in E) \wedge ((b, r) \in E)$, with E the set of non oriented edges of the initial graph and V the set of vertices of the initial graph.

Then the function to identify two-neighbours vertices is:

```
for all( A in V ) {
    for all ( (A,B) in E ) {
        for all ( (B,C) in E ) {
            assert(neighbours(A,C))
        }
    }
}
```

With the definition of two-neighbours, clauses can now be written.

$\forall a, b \in V$ such that a and b are not two-neighbours, $\forall y \in V'$, $cl(\neg \mathbf{m}_{a,y} \vee \neg \mathbf{m}_{b,y})$.

These clauses can easily be translated with the pseudocode:

```
for all( A in V and B in V ) {
    if ( not(neighbours(A,B)) ) {
        for all( Y in V' ) {
            write(-m(a,y) -m(b,y))
        }
    }
}
```

Strict two-neighbours restriction is much faster and simpler than good-path merge restriction. An evaluation on BioModels will be performed in next section.

5.3. Evaluation

The strict merge restriction was first evaluated on handmade models and then tested on bigger models from BioModels.

Handmade models

Table 5-3 and table 5-4 compare the number of SEPIs with the initial SEPI framework (line "Without") and with the strict merge restriction (line "Strict"). Table 5-3 presents results on combinations of two Michaelis-Menten patterns. It can be observed that strict merge restriction permits to reduce the number of unwanted reductions. Table 5-4 presents results on MAPK cascades.

With only three models for three-levels MAPK cascade in the three reduction forms, no difference is observed with or without the strict merge restriction when the number of SEPIs is recorded until 200.

With the set of 15 models corresponding to MAPK cascades with 1, 2 or 3 levels and the three different forms of reduction, merge restriction permits to eliminate the timeouts and reduce the number of reductions.

Restriction	Pairs with SEPIs ≥ 200	Pairs with SEPIs < 200	Number of SEPIs mean	Number of SEPIs median	No SEPI
Expanded and intermediary					
Without	0	158 (07.60%)	1.65	2	1912 (92.40%)
Strict	0	23 (01.00%)	1.6	2	2047 (99.00%)
Intermediary and Reduced					
Without	7 (00.30%)	225 (10.90%)	17.5	4	1838 (88.80%)
Strict	0	76 (03.70%)	10.3	6	1994 (96.30%)
expanded and Reduced					
Without	92 (04.40%)	260 (12.60%)	25	4	1718 (83.00%)
Strict	19 (00.90%)	173 (08.40%)	29	18	1878 (90.70%)
Total					
Without	99 (01.60%)	643 (10.40%)	16.4	2	5468 (88.00%)
Strict	19 (00.30%)	272 (04.40%)	21.4	10	5919 (95.30%)

Table 5-3.: SEPIs for combinations of Michaelis-Menten.

Reduction	Pairs with SEPIs ≥ 200	Pairs with SEPIs < 200	Number of SEPIs mean	Number of SEPIs median	No SEPI	Timeouts
Without	64 (30.50%)	9 (04.00%)	25	14	127 (60.50%)	10 (05.00%)
Strict	54 (26.00%)	21 (10.00%)	60	39	135 (64.00%)	0

Table 5-4.: SEPIs for MAPK cascades.

BioModels

An evaluation on models of BioModels was also made.

Figures 5-3 to 5-6 are a graphical presentation of the results. In all four figures, yellow vertices represent MAPK models, green vertices represent models of the Circadian Clock class, blue vertices represent Cell Cycle models and purple vertices represent models of the Calcium Oscillation class.

Figure 5-3 shows SEPI relations intra class with the initial SEPI framework. Figure 5-4 shows SEPI relations intra class after the implementation of the strict merge restriction. Numbers on edges represent the amount of SEPIs found between the two models.

When a SEPI relation is found between two models, it can be observed in the figures that the size of the set of SEPIs is decreasing. But a diminution of SEPI relations can also be observed. Table 5-5 will give more precise results.

Figure 5-5 shows SEPI relations inter class with the initial SEPI framework. Figure 5-6 shows SEPI relations inter class after the implementation of the strict merge restriction.

A significant diminution of unwanted SEPIs inter class can be observed in the figures. Quantitative results will be displayed in table 5-6.

Table 5-5 and table 5-6 present more precise results. Each block of two lines present results of a specific class or for the comparison of two specific classes. For each class, or pair of classes, the number in brackets is the number of tested pairs of models. For each class, or pair of classes, two lines of results are presented, the line "Without" presents the number of SEPIs with the initial SEPI framework, the line "Strict" presents the number of SEPIs with the strict merge

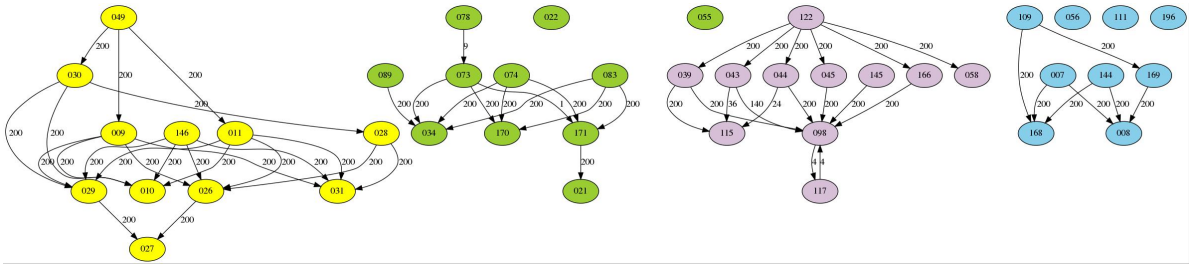

Figure 5-3.: SEPIs intra class without merge restriction.

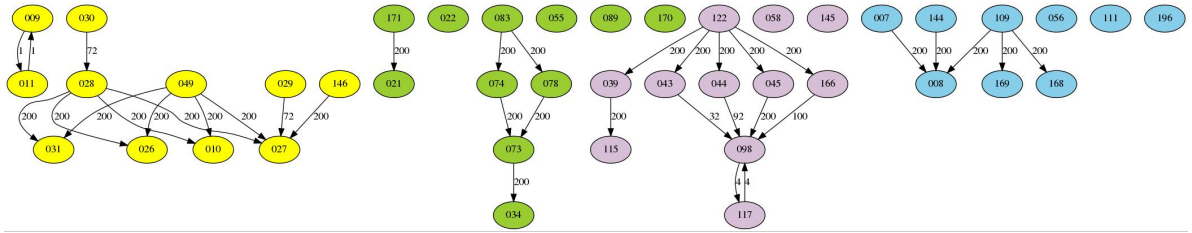

Figure 5-4.: SEPIs intra class with strict merge restriction.

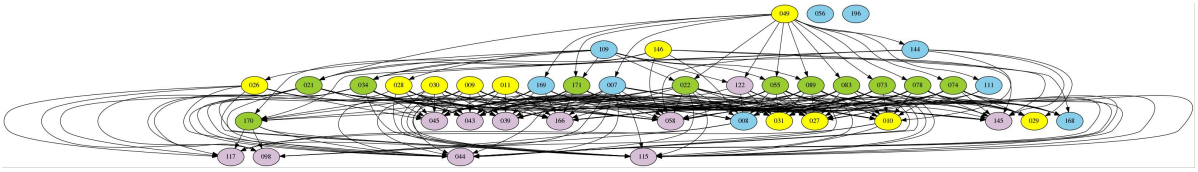

Figure 5-5.: SEPIs inter class without merge restriction.

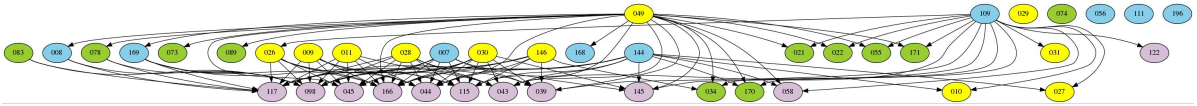

Figure 5-6.: SEPIs inter class with strict merge restriction.

restriction. For each line, the number of SEPIs, the number of no SEPI relations and the number of Timeouts are given. For SEPIs intra class, the size of the set of SEPIs is also relevant, the number of set with a size bellow 200 is also specified.

Several observations can be made from the two tables:

- The strict merge restriction eliminates 64.49% of unwanted SEPIs inter class.
- The strict merge restriction eliminates 80.23% of timeouts inter class.
- The strict merge restriction increases by a factor of 1.33 the number of set of SEPIs bellow a size of 200 among SEPIs intra class.
- The strict merge restriction eliminates 78.05% of timeouts intra class.
- The strict merge restriction makes 42.05% of SEPI intra class disappear.

Those observations are the targeted results except for the last one. But overall, considering the low complexity of this implementation, those results are encouraging.

Class	Restriction	SEPI ≥ 200	SEPI < 200	SEPI total	No SEPI	Timeout (20 min)
Ca (110)	Without	20	6	26 (23.64%)	83 (75.45%)	1 (00.91%)
	Strict	11	8	19 (17.27%)	91 (82.73%)	0 (00.00%)
Cell (42)	Without	8	0	8 (19.05%)	28 (66.67%)	6 (14.28%)
	Strict	5	0	5 (11.90%)	34 (80.95%)	3 (07.15%)
Circ (110)	Without	16	3	19 (17.27%)	70 (63.64%)	21 (19.09%)
	Strict	10	0	10 (09.09%)	100 (90.91%)	0 (00.00%)
MAPK (110)	Without	35	0	35 (31,82%)	62 (56,36%)	13 (11,82%)
	Strict	13	4	17 (15.45%)	87 (79.09%)	6 (05.46%)
Total (372)	Without	79	9	88 (23,66%)	243 (65,32%)	41 (11,02%)
	Strict	39	12	51 (13.71%)	312 (83.87%)	9 (02.42%)

Table 5-5.: SEPI relations intra class with and without strict merge restriction.

Pair of classes			Restriction	SEPI	No SEPI	Timeout
Ca	Cell	(154)	Without	43 (27.92%)	109 (70.78%)	2 (01.30%)
			Strict	33 (21.43%)	121 (78.57%)	0 (00.00%)
Ca	Circ	(242)	Without	100 (41.32%)	136 (56.20%)	6 (02.48%)
			Strict	4 (01.65%)	238 (98.65%)	0 (00.00%)
Ca	MAPK	(242)	Without	74 (30.58%)	162 (66.94%)	6 (02.48%)
			Strict	51 (21.08%)	190 (78.51%)	1 (0.41%)
Cell	Circ	(154)	Without	22 (14.29%)	109 (70.78%)	23 (14.93%)
			Strict	7 (04.55%)	142 (92.21%)	5 (03.24%)
Cell	MAPK	(154)	Without	30 (19.48%)	105 (68.18%)	19 (12.34%)
			Strict	9 (05.84%)	135 (87.66%)	10 (06.50%)
Circ	MAPK	(242)	Without	52 (21.49%)	155 (64.05%)	35 (14.46%)
			Strict	10 (04.13%)	230 (95.04%)	2 (00.83%)
Total		(1188)	Without	321 (27.02%)	776 (65.32%)	91 (7.66%)
			Strict	114 (09.60%)	1056 (88.89%)	18 (01.51%)

Table 5-6.: SEPI relations inter class with and without strict merge restriction.

Model	No Restriction													Merge Restriction											
		Normal				Min ⊥				Max ⊥					Normal				Min ⊥				Max ⊥			
	time	nb	1st	2nd	last	nb	1st	2nd	last	nb	1st	2nd	last	time	nb	1st	2nd	last	nb	1st	2nd	last	nb	1st	2nd	last
122-098 (Ca)	26	200	10	11	1406	200	4	4	7	200	109747	98568	146762	13	200	25	19	1881	160	1448	1377	2986	200	623	632	933
043-117 (Ca)	5	140	4	4	47	28	2	1	1	16	2	2	3	1	32	5	4	7	16	2	1	3	16	2	1	3
144-008 (Cell)	591	200	97	93	316	200	63	106	289	-	t	-	-	122	200	836	1101	1033	200	186	203	420	30	198	210	271
007-168 (Cell)	687	200	200	159	360	-	t	-	-	-	t	-	-	91	0	(570)	-	-	0	(58)	-	-	0	(56)	-	-
021-170 (Circ)	164	200	7316	1282	7775	200	1186	536	396	200	2017	1411	2216	34	0	(76)	-	-	0	(9)	-	-	0	(9)	-	-
083-034 (Circ)	873	200	333	185	531	200	62852	61760	63432	-	t	-	-	215	200	528	488	835	200	832	806	1528	200	805	651	731
029-027 (MAPK)	26	72	12	12	29	1	3	(2)	-	4	3	3	3	1	72	12	12	30	1	3	(3)	-	4	3	3	3
011-026 (MAPK)	485	200	45258	45204	48402	200	75755	74830	85844	-	t	-	-	64	0	(469)	-	-	0	(46)	-	-	0	(50)	-	-
mapk1-mapk2	3193	1	t	-	-	1	t	-	-	1	t	-	-	98	1	1677	(3679)	-	1	426	(374)	-	1	411	(373)	-
mapk1-mapk3	796	200	t	-	-	200	144562	146264	150344	-	t	-	-	55	200	607	603	3427	165	459	539	835	64	456	456	504
mapk2-mapk3	460	200	t	-	-	1	988840	(988642)	-	-	t	-	-	33	200	234	216	2729	16	77	119	120	4	94	98	98

Table 5-7.: Time to compute the first second and last SEPI for each problem (time in ms).

Execution time

Execution time of the new framework is also an important criterion for biologists.

Table 5-7 shows the time needed to write clauses and to solve specific instances.

To make a benchmark, two models of each class were selected. For the benchmark to be representative, small and big models were selected. Handmade models of MAPK cascade were also used as comparison, they are represented in the table by "mapk1", "mapk2" and "mapk3".

The first column of the table (called *Model*) represents each problem. For each problem, a SEPI relation is searched between two models of the same class, the corresponding class is specified in between brackets.

Is the first half of the table (called *No Restriction*), tests were made without any merge restriction. In the second half of the table (called *Merge Restriction*), the strict two-neighbours merge restriction was used.

The column *time* specifies the time needed to write the clauses. In column *time* of the *No Restriction*'s half, it corresponds to the time for all clauses of the SEPI framework. In column *time* of *Merge Restriction*'s half, it's only the additional time needed to write the clauses corresponding to the merge restriction. All times of the table are in milliseconds.

Then for each half (*no restriction* or *merge restriction*) and for each problem (each pair of models) a SEPI relation is searched without any extremalisation of the number of vertex deletions (column *Normal*) then with minimisation of the vertex deletion number (column *Min* $\perp$) and with maximisation of the vertex deletion number (column *Max* $\perp$).

When a SEPI relation is searched between two models the column *nb* corresponds to the number of different SEPI relations found. Because there can be thousands of SEPI relation, only the first 200 ones were computed. When the set of SEPI relations exceed 200 it is not humanly possible to look for an interpretation of each of them so the goal is to keep this number below 200. Thus *200* in column *nb* represents 200 or more SEPI relations.

For each pair of models and for each problem, columns *1st*, *2nd* and *last* represent respectively the time needed to find the first the second and the last SEPI. A *t* represents a timeout. Timeouts are declared after running the SAT solver for 20 minutes without result. A number in brackets represents the time needed to establish that the problem was not satisfiable (no SEPI). A - represents the absence of result (no time for second and last SEPI when there exists only one SEPI and no number of SEPIs when there is a timeout).

A few observations can be made about the results of this table:

- It takes a relatively low amount of time to write the clauses corresponding to the two-neighbours merge restriction.
- It is more complex for a MAX-SAT solver to maximise the number of deletion than to minimise it.
- The two-neighbours merge restriction makes all timeouts disappear.
- Maximisation of the number of deletions in addition to the two-neighbours merge restriction is very efficient, especially for handwritten models of MAPK.
- The two-neighbours merge restriction makes three SEPI relations disappear (for pair of models 007-168, 021-170 and 011-026).

5.4. Conclusion

The strict merge restriction is better than the accurate merge restriction in terms of computational complexity. Results are obtained faster, the strict merge restriction reduces significantly the number of timeouts during tests of models of BioModels.

This restriction is stronger and makes some wanted SEPI relations disappear but it also reduces the amount of unwanted SEPIs inter class.

This is a good evolution of the SEPI framework and it is now implemented and accessible for users of Biocham. But results given my the SEPI framework with strict merge restriction need to be taken carefully, a SEPI of a complex form can still exist between big models and not been shown by the updated version of the framework.

6. Pattern reduction

6.1. Motivations

In previous sections, two main improvements were made on the SEPI framework:

. Extremalisation of the vertex deletion number has been implemented (section 3) in order to reduce the size of the set of SEPIs found between two given graphs.

. Restrictions on the merge definition were implemented (section 5) in order to reduce the number of non biologically explainable SEPIs.

These improvements showed encouraging results, for example:

- 76.32% of SEPI sets were above a size of 200 units in the class Calcium Oscillations before minimisation of the number of vertex deletions and only 42,11% after.
- 27.02% SEPIs were found between models of different classes before implementing the merge restriction, only 9.60% after.
- 11.02% of model pairs are too complex for the SEPI framework to compute them in a reasonable time. Only 2.42% are too complex for the SEPI framework with the merge restriction.

But previous approaches still allow some unexpected pairings, as shown in figure 6-1 and figure 6-2. A SEPI with two-neighbours merge restriction and minimisation of the number of deletions is searched from the initial graph to the image graph.

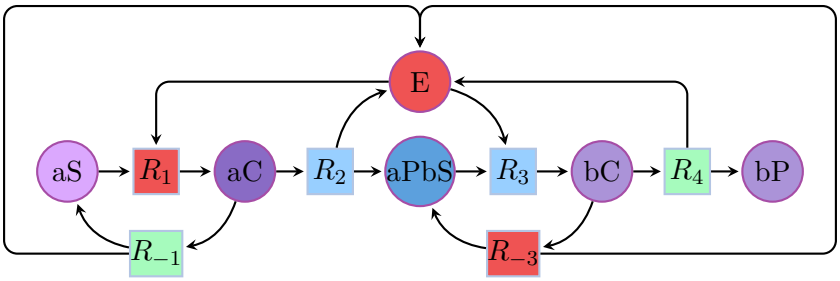

Figure 6-1.: Unexpected SEPI: initial graph.

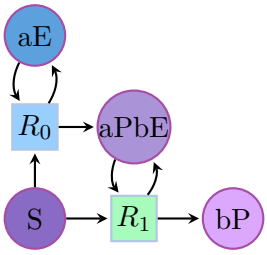

Figure 6-2.: Unexpected SEPI: image graph.

Figure 6-1 is the initial graph: it represents a simple combination of two Michaelis-Menten reactions with complexes substrate enzyme and with reverse reactions R_{-1} and R_{-3}. The first product is the substrate of the second reaction and the enzyme is shared. This combination is quite natural because it can be observed in models of MAPK.

Figure 6-2 is the image graph: it represents a chain of two Michaelis-Menten, both in reduced form, where the first product acts as an enzyme on the same substrate to produce a second product.

A SEPI μ with strict two-neighbours merge restriction and deletion number minimisation is found between the two graphs. μ is represented by the colours in the figures. Vertices of one colour in figure 6-1 are mapped to the vertex of the same colour in figure 6-2:

$\mu(aPbS) = aE$,

$\mu(aC) = S$,

$\mu(R_2) = \mu(R_3) = R_0$, mapping those two reactions on the same vertex is possible because $aPbS$ is a common neighbour,

$\mu(bC) = \mu(bP) = aPbE$, mapping those two reactions on the same vertex is possible because R_4 is a common neighbour,

$\mu(R_{-1}) = \mu(R_4) = R_0$, mapping those two reactions on the same vertex is possible because E is a common neighbour,

$\mu(aS) = bP$,

$\mu(R_1) = \mu(R_{-3}) = \mu(E) = \perp$.

This SEPI does not have a biological interpretation but was not filtered by the strict two-neighbours merge restriction because the shared enzyme makes all reactions two-neighbours.

This pairing is not relevant because, for example, the entry aS of the first expended Michaelis-Menten pattern becomes the output bP of the second reduced Michaelis-Menten pattern and the complex aC of the first expended Michaelis-Menten pattern becomes the entry S of the second reduced Michaelis-Menten pattern.

Some patterns like the Michaelis-Menten pattern were precisely studied. The expected reduction of those patterns are well known. Thus, another way of filtering more SEPIs without biological interpretation is to reduced known expanded patterns.

Reducing patterns before using the SEPI framework between two given graphs will also reduce the size of the set of SEPIs found between those two graphs.

Table 6-1 is a good example. It represents all possible reductions between the complete Michaelis-Menten reaction graph represented in figure 6-3 and the reduced Michaelis-Menten reaction graph represented in figure 6-4. One line represents a SEPI. One column represents one vertex in the initial graph and it's values in the image graph through each SEPIs. It can already be observed that there is three different SEPIs just for the reduction of a Michaelis-Menten reaction.

Sepi	E	S	ES	P	R_1	R_{-1}	R_2
1	E	S	E	P	R	$\perp$	R
2	E	S	$\perp$	P	R	$\perp$	R
3	$\perp$	S	E	P	R	$\perp$	R

Table 6-1.: All possible reductions between complete and reduced Michaelis-Menten reaction graphs.

Furthermore, reducing patterns before using the SEPI framework will reduce the computation time of the subgraph isomorphism research as it will reduce the number of vertices. And reduced graphs are also easier to study for biologists.

The graph rewriting strategy is the following:

1. Search known patterns in G_1 and G_2,
2. Rewrite graphs to obtain graphs G_1' and G_2' by replacing expanded patterns by their reduced form,
3. Search SEPIs between G_1' and G_2'.

This pre-processing is expected to reduce both the number of pairs that present a SEPI by eliminating SEPIs without biological interpretation, and to reduce the number of SEPIs for pairs of graphs that have at least one SEPI with interpretation.

Reducing pattern is a variant of the subgraph isomorphism (SISO) problem. But searching SISOs is not sufficient in this case because there is more constraints on a subset of vertices. For example in the Michaelis-Menten pattern, the complex ES in figure 6-3 cannot interfere in any other reactions and reactions R_1, R_{-1} and R_2 cannot interfere with other species.

This section will present the graph rewriting strategy, also named constrained subgraph isomorphism (cSISO) problem. In part 6.2, a broader framework of cSISO will be presented. This framework allow the search of any pattern with constraints on a subset of vertices. Although part 6.4 will show that the implementation is only made for patterns defined in part 6.3.

6.2. Definitions

The Michaelis-Menten pattern will be the main example of this section because it is a well studied reaction but other patterns with known reductions can also be used. The graph in figure 6-3 is an example of reaction graph with expended Michaelis-Menten pattern. The graph in figure 6-4 is the equivalent of the previous reaction graph with reduction of the Michaelis-Menten pattern.

Only searching subgraph isomorphisms is not sufficient in this pattern reduction because there are more constraints :

- the complex ES cannot interfere in other reactions,
- reactions R_1, R_{-1} and R_2 cannot use other species.

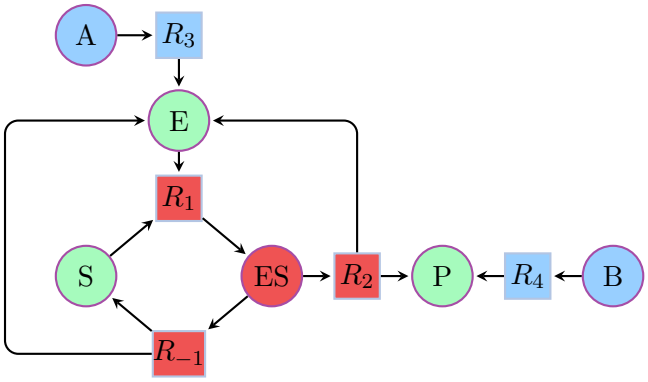

Figure 6-3.: Reaction graph with expended Michaelis-Menten pattern.

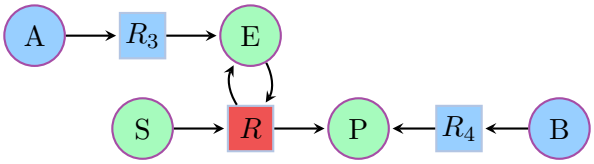

Figure 6-4.: Reaction graph with reduced Michaelis-Menten pattern.

Isolated vertices with edge constraints are circled in red in figures. Non isolated vertices without constraints are circled in green in figures. Vertices that are not part of the pattern are circled in blue in figures.

In this section, theory is given for general graphs (not necessarily bipartite), but it extends immediately to reaction graphs.

Let $G_1 = (V_1, E_1)$ be a graph with V_1 a set of vertices and E_1 a set of arcs. Let F_1 be a subset of V_1 corresponding to fixed vertices. For example, in the reaction graph with expended Michaelis-Menten pattern of figure 6-3, $F_1 = \{ES, R_1, R_{-1}, R_2\}$.

Let $G_2 = (V_2, E_2)$ be a second graph defined in the same way, with F_2 a subset of V_2.

A pattern defined by a graph G_1 is searched in a second graph G_2 with constraints defined by F_1. If $F_1 = \emptyset$, the constrained subgraph isomorphism problem corresponds to searching an induced subgraph isomorphism.

The definition is given with a possible fixed set F_2 to get a partial order on the space of pairs (G, F) where G is a graph and F a subset of vertices of G.

Definition 6.2.1 (Constrained subgraph isomorphism) *A constrained subgraph isomorphism (cSISO) from G_1 to G_2 constrained by F_1 and F_2 is a function $\mu : G_1 \to G_2$ such that:*

1. *$\mu(G_1)$ is an induced subgraph of G_2,*
2. *μ is a graph isomorphism from G_1 to $\mu(G_1)$,*
3. *$\mu(F_1) \subset F_2$, and*
4. *for all $u \in F_1$, for all $z \in V_2$, $(\mu(u), z) \in E_2 \implies z \in \mu(G_1)$ (all neighbours of $\mu(u)$ are images of neighbours of u)*

This definition extends easily to bipartite directed graphs.

$(G_1, F_1) \overset{cSISO}{\rightsquigarrow} (G_2, F_2)$ is written when there exists a cSISO from (G_1, F_1) to (G_2, F_2).

Property 6.2.1 *The partial order given by $\overset{cSISO}{\rightsquigarrow}$ is not a well quasi-order.*

Proof: The set $\{(C_n, \emptyset), n \geq 3\}$, where C_n is the cycle graph with n vertices, is an infinite antichain. □

Definition 6.2.2 (Constrained subgraph isomorphism problem.) *The constrained subgraph isomorphism problem is the decision problem:*

Instance: Two graphs and associated subsets of vertices $(G_1, F_1), (G_2, F_2)$

Question: $(G_1, F_1) \overset{cSISO}{\rightsquigarrow} (G_2, F_2)$?

Theorem 6.2.1 *The constrained subgraph isomorphism problem is NP-complete.*

In the case $F_1 = F_2 = \emptyset$, the cSISO becomes the induced subgraph isomorphism problem. The induced SISO problem is known to be NP-complete as the search of a k-clique is NP-complete and can be reduced to SAT problem [37].

In the case of a pattern reduction in reaction graphs, (G_1, F_1) is fixed and $F_2 = V_2$. The time and space complexities will therefore be linear in $|V(G_2)|$, event to find all the cSISOs from (G_1, F_1) to (G_2, F_2).

The reduction of patterns is not always commutative, it depends of the pattern.

This definition of cSISO is general but pattern reduction cannot be easily implemented with every patterns. A few characteristics need to be respected:

- for the commutative property, patterns cannot overlap in their extended forms,
- vertices of F_1 form a connected component in G_1.

For one cSISO $\mu : (G_1, F_1) \to (G_2, F_2)$ and a reduction $r : G_1 \to G_1'$, let denote $r_\mu(G_2)$ the reduced graph obtained from G_2 by reducing the pattern $\mu(G_1)$.

Let assume that the reduction r preserves vertices in $V(G_1) \backslash F_1$ and modify (delete or merge) vertices in F_1 (in fact, this is how we choose the set F_1). Then, relevant patterns G_1 and their reduction r verify : for any G_2 and cSISOs $\mu, \nu : (G_1, F_1) \to (G_2, V(G_2))$, either $r_\mu(G_2) = r_\nu(G_2)$, either $\mu(G_1)$ is still a subgraph of $r_\nu(G_2)$ and we can apply r_μ to $r_\nu(G_2)$. This last requirement is equivalent to $\nu(F_1) \cap \mu(G_1) = \emptyset$.

6.3. Patterns

In this part will be presented different patterns that will be reduced in the same set of reaction graphs used previously.

A necessary condition for pattern reduction to be commutative is that the vertices of F_1 form a connected component in G_1. All patterns presented respect this condition.

In all reaction graphs of this section isolated vertices with edge constraints are circled in red in figures and non isolated vertices without constraints are circled in green in figures.

The Michaelis-Menten pattern and its reduction were already shown in figure 6-3 and figure 6-4.

The distributive Michaelis-Menten pattern shown in figure 6-5 will also be reduced. Its reduced form it also the Michaelis-Menten reduced form from figure 6-4. This distributive kinetics used for the dephosphorylation in a MAPK cascade was studied by Markevich in 2004 [38]. It's a model for the dephosphorylation: $E + S \Leftrightarrow ES \rightarrow EP \Leftrightarrow E + P$.

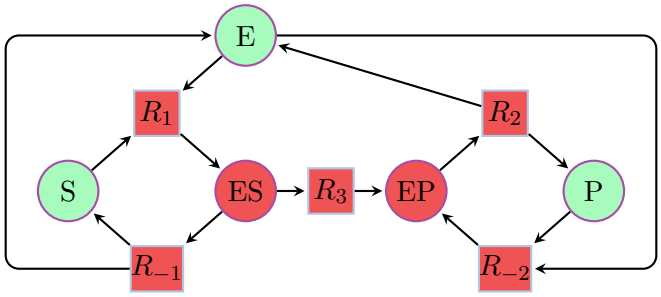

Figure 6-5.: Distributive Michaelis-Menten reaction graph.

The Hill pattern with two distinct binding sites shown in figure 6-6 will also be reduced to the Michaelis-Menten reduced form from figure 6-4. This pattern was studied by Moreland et al. [39] it represents a two-steps enzymatic mechanism with two binding sites.

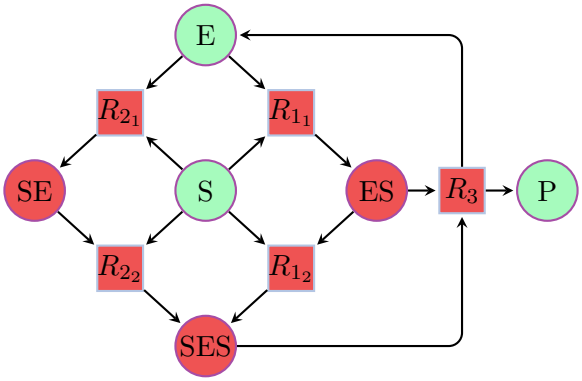

Figure 6-6.: Hill reaction graph.

The partial Hill pattern with twice the same binding site shown in figure 6-7 will also be reduced to the Michaelis-Menten reduced form from figure 6-4. This pattern was studied by Good et al. [40] it is a variant of the Hill reaction.

The double Michaelis-Menten pattern with two forms of the enzyme shown in figure 6-8 will also be reduced to the Michaelis-Menten reduced form from figure 6-4.

6.4. Implementation

Strategy

Pattern reduction is a preprocessing step before searching SEPIs between two graphs.

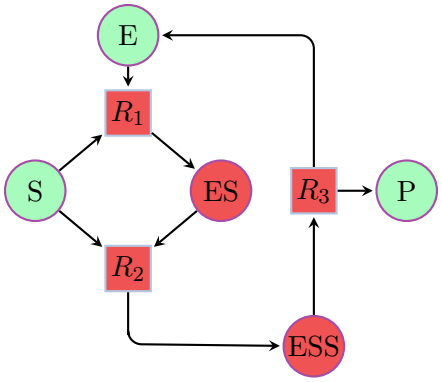

Figure 6-7.: Partial Hill reaction graph.

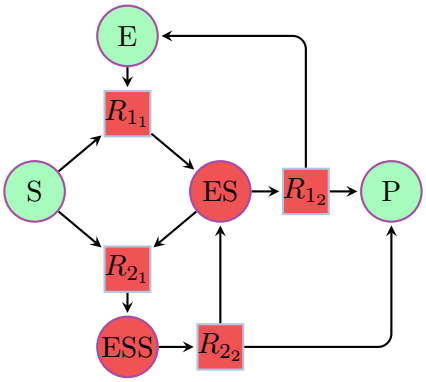

Figure 6-8.: Double Michaelis-Menten reaction graph.

The graph rewriting strategy is the following:

1. Search known patterns in G_1 and G_2,
2. Rewrite graphs to obtain graphs G_1' and G_2' by replacing expanded patterns by their reduced form,
3. Search SEPIs between G_1' and G_2'.

Data structure

The data structure used in the previous SEPI framework was not performant enough and the implementation would have been with a high complexity.

With the previous data structure a graph was represented by a number of vertices, a number of species and a list of pairs representing edges:

```
Graph = [|V|, |S|, list_edges]
```

With V a set of vertices and |V| it's cardinality and S a set of species vertices and |S| it's cardinality. Vertices were represented by integers and ordered. The set of specie vertices is $[\![0, |S| - 1]\!]$. The set of reaction vertices is $[\![|S|, |V| - 1]\!]$.

With this data structure, the Michaelis-Menten pattern reduction problem would have a complexity of $\mathcal{O}(|V|^7)$: the existence of a Michaelis-Menten pattern need to be checked for each vertex of the graph.

A better solution was to change the data structure by constructing two dictionaries, *ingoing_edges* and *outgoing_edges*.

$\forall v \in V$, $ingoing\_edges[v] = \{u | (u, v) \in list\_edges\}$

$\forall v \in V$, $outgoing\_edges[v] = \{u | (v, u) \in list\_edges\}$

With this new data structure, the complexity for every pattern is only $\mathcal{O}(|V|)$, as it will be shown in the pseudocode subsection.

Pseudocode

This strategy does not need a SAT solver, all the encoding is made in Prolog C.

The Michaelis-Menten pattern will be taken as example for the implementation. Checking if a vertex is the core vertex of a Michaelis-Menten pattern has a constant cost. The core vertex of the pattern is the complex ES. To check the pattern, it is sufficient to check all ongoing and outgoing edges of isolated vertices. Is the Michaelis-Menten pattern, ES, R_1, R_{-1} and R_2 are fixed vertices. The function *michaelis_menten_pattern* checks ongoing and outgoing edges and identifies species E, S and P that will be needed for the graph rewriting part.

```
michaelis_menten_pattern(+ES, -E, -S, -P):-
  ingoing_edges[ES]   = [R1],
  outgoing_edges[ES]  = [R_1, R2],
  ingoing_edges[R1]   = [E, S],
  outgoing_edges[R1]  = [ES],
  ingoing_edges[R_1]  = [ES],
  outgoing_edges[R_1] = [E, S],
  ingoing_edges[R2]   = [ES],
  outgoing_edges[R2]  = [E, P].
```

To identify all patterns of a graph, it is then sufficient to review all specie vertices to check if they are the core of the pattern.

```
for all( Vertex in V ) {
    if ( michaelis_menten_pattern(+Vertex, -E, -S, -P) ):
        rewrite_reaction("E+S=>E+P.")
}
```

If they are the core of the pattern, reactions R_1, R_{-1} and R_2 can all be rewritten by a simple reaction R: $E + S => E + P$.

After reviewing all vertices of the initial graph, all reactions that were not in the pattern can be added to the new graph.

6.5. Evaluation

The first implementation of the pattern reduction function was made only for variants of the Michaelis-Menten pattern.

Figure 6-9 presents all possible Michaelis-Menten patterns. A Michaelis-Menten pattern is always composed of the three non isolated species E, S and P and at least of three isolated vertices, ES, R_1 and R_2. The reaction R_{-1} can be present or not. The reaction R_{-2} is rarely present. The specie EP and the reaction R_3 are always present together but also rarely present.

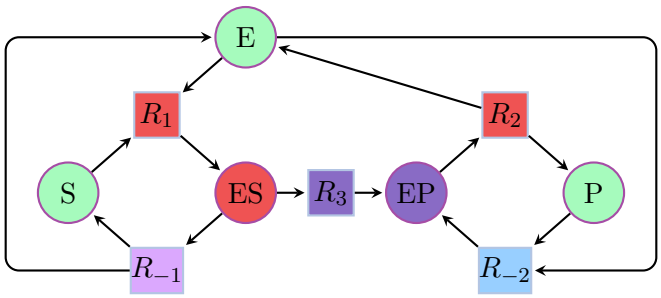

Figure 6-9.: All variants of the Michaelis-Menten pattern.

All those variants of the Michaelis-Menten pattern were searched in all models of the Calcium Oscillations class (Ca), the Cell Cycle class (Cell), the Circadian Clock class (Circ) and the MAPK class (MAPK).

Results are presented in table 6-2. Theses models are coming from BioModels and they are each represented by a number, which correspond to column "Model" in the table. Classes Ca, Circ and MAPK are composed of eleven models, Cell is composed of seven models. Models appearing in this table are only models where at least one variant of Michaelis-Menten pattern was found. No patterns were found in the other models. Column "Without R-1" represents the smallest variant of the Michaelis-Menten pattern with only R_1, ES and R_2 as isolated vertices (red and green vertices in figure 6-9). Column "With R-1" represents the pattern with the smallest variant of Michaelis-Menten and the reaction R_{-1} (red, green and pink vertices in figure 6-9). Column "With R-1, R-2, EP" represent the biggest variant of the Michaelis-Menten pattern (all vertices in figure 6-9). Other variants of the Michaelis-Menten pattern, reaction R_{-2} without reaction R_{-1} for example, were not found in the selected models.

Class		Model	With R-1	Without R-1	With R-1, R-2, EP
Ca	(11)	39	0	1	0
Cell	(7)	109	8	0	0
		169	0	1	0
Circ	(11)	-	-	-	-
MAPK	(11)	9	10	0	0
		11	10	0	0
		26	2	0	2
		28	4	0	3
		30	4	0	4
		49	13	0	0

Table 6-2.: Number of Michaelis-Menten patterns.

It can be observed in table 6-2 that only three models outside of the MAPK class were concerned by Michaelis-Menten patterns. Thus after reducing all variants of the Michaelis-Menten pattern, tests with the SEPI framework were only conducted on the MAPK class. Tests were performed with the initial SEPI framework, without merge restrictions and without extremalisation of the number of vertex deletion. Results can be observed in table 6-3.

	SEPI			No SEPI	Timeout
	≥ 200	< 200	total		
Normal	35	0	35 (31,82%)	62 (56,36%)	13 (11,82%)
Reduced	22	6	28 (25.45%)	81 (73.64%)	1 (0.91%)

Table 6-3.: SEPIs in MAPK class.

Table 6-3 presents results for the 110 pairs of models of the MAPK class. Figures 6-10 and 6-11 also represent the results.

Three observations can be made from the table and the figures:

- The number of set of SEPIs which size is bellow 200 goes from zero to six. Reducing this size was the goal of pattern reduction.

- The number of timeout of considerably decreasing which is a very encouraging result.

- The number of total SEPIs is decreasing, which can be concerning. In fact, models 27, 29 and 31 of the MAPK class are variant of respectively models 26, 28 and 30. Markevich et al. also tried to reduce patterns but didn't use the same methodology, that's why, after reducing Michaelis-Menten patterns, some SEPIs with models 27, 29 and 31 cannot be found any more.

The other patterns, Hill reaction, partial Hill reaction and double Michaelis-Menten, were also implemented but were not found among models of the four classes.

6.6. Evaluation of all methods combined

Table 6-4 and table 6-5 present results of the updated SEPI framework with the three improvements: maximisation of the vertex deletion number, strict two-neighbours merge restriction and reduction of Michaelis-Menten patterns.

A few observations can be made from these two tables:

- The number of timeouts is noticeably decreasing both between models of different classes and between models of the same class.

- The number of SEPIs inter class is reduced considerably. Especially with models of the class Circadian Clock.

- The size of the set of SEPIs between two given models of the same class is also decreasing.

- The number of SEPIs intra class is also reduced by the combined methods.

Class	Restriction	SEPI ≥ 200	SEPI < 200	SEPI total	No SEPI	Timeout (20 min)
Ca (110)	Without	20	6	26 (23.64%)	83 (75.45%)	1 (00.91%)
	Combined	7	10	17 (15.45%)	93 (84.55%)	0
Cell (42)	Without	8	0	8 (19.05%)	28 (66.67%)	6 (14.28%)
	Combined	1	0	1 (02.38%)	35 (83.33%)	6 (14.29%)
Circ (110)	Without	16	3	19 (17.27%)	70 (63.64%)	21 (19.09%)
	Combined	0	1	1 (00.91%)	105 (95.45%)	4 (03.64%)
MAPK (110)	Without	35	0	35 (31,82%)	62 (56,36%)	13 (11,82%)
	Combined	1	10	11 (10.00%)	89 (80.91%)	10 (09.09%)
Total (372)	Without	79	9	88 (23,66%)	243 (65,32%)	41 (11,02%)
	Combined	9	21	30 (08.06%)	322 (86.56%)	20 (05.38%)

Table 6-4.: SEPI relations intra class with combined restrictions.

Pair of classes			Restriction	SEPI	No SEPI	Timeout
Ca	Cell	(154)	Without	43 (27.92%)	109 (70.78%)	2 (01.30%)
			Combined	16 (10.49%)	126 (81.82%)	12 (07.69%)
Ca	Circ	(242)	Without	100 (41.32%)	136 (56.20%)	6 (02.48%)
			Combined	12 (04,96%)	230 (95.04%)	0
Ca	MAPK	(242)	Without	74 (30.58%)	162 (66.94%)	6 (02.48%)
			Combined	5 (02.07%)	226 (93.39%)	11 (04.54%)
Cell	Circ	(154)	Without	22 (14.29%)	109 (70.78%)	23 (14.93%)
			Combined	0	144 (93,51%)	10 (06,49%)
Cell	MAPK	(154)	Without	30 (19.48%)	105 (68.18%)	19 (12.34%)
			Combined	4 (02.60%)	133 (86.36%)	17 (11.04%)
Circ	MAPK	(242)	Without	52 (21.49%)	155 (64.05%)	35 (14.46%)
			Combined	0	231 (95.45%)	11 (04.55%)
Total		(1188)	Without	321 (27.02%)	776 (65.32%)	91 (07.66%)
			Combined	37 (3.11%)	1090 (91.75%)	61 (5.14%)

Table 6-5.: SEPI relations inter class with combined restrictions.

Furthermore, it was noticed in a lot of SEPI sets that species had, most of the time, the same image through SEPI reductions and only reactions had distinct images. However, biologists are more interested by images of species rather than images of reactions.

Another small SEPI filter was implemented to list only SEPIs with distinct species images. Results can be observed in table 6-6. This filter was tested alone (lines *Initial f. distinct s.*) and combined with the strict two neighbours merge restriction and maximisation of the number of vertex deletion (line *Combined m. distinct s.*). Previous results are displayed in the table for comparison.

There is a noteworthy diminution of the SEPI number between two given models, especially for models of the class Calcium Oscillations.

Class	Filter	≥ 200	< 200	mean
Ca	Initial framework	20	3	58.00
	Initial f. distinct s.	8	15	7.33
Ca	Combined methods	7	10	19.20
	Combined m. distinct s.	1	16	7.19
Cell	Initial framework	8	0	-
	Initial f. distinct s.	7	1	1.00
Cell	Combined methods	1	0	-
	Combined m. distinct s.	0	1	15.00
Circ	Initial framework	16	3	13.67
	Initial f. distinct s.	16	3	1.67
Circ	Combined methods	0	1	6.00
	Combined m. distinct s.	0	1	2.00
MAPK	Initial framework	35	0	-
	Initial f. distinct s.	34	1	12
MAPK	Combined methods	1	10	29.60
	Combined m. distinct s.	0	11	18.36
Total	Initial framework	79	6	35.83
	Initial f. distinct s.	66	19	6.39
Total	Combined methods	9	21	23.52
	Combined m. distinct s.	1	29	11.52

Table 6-6.: SEPI relations intra class.

6.7. Conclusion

The pattern reduction strategy shows good results with both the diminution of the number of timeout for the SEPI framework and a diminution of the size of the set of SEPIs between two given models. But these good results are only in the class MAPK. Other patterns and Michaelis-Menten patterns were not found in other classes.

Implementing the pattern reduction strategy need a lot of knowledge of the patterns and their reduction and the initial goal of the SEPI framework was to be able to search reductions without knowledge of the model and by using only the reaction graph. Maybe other patterns can be found in the models of BioModels but it would require a deeper analysis.

These results are non the less encouraging and the pattern reduction function is implemented in Biocham. It is now available for users to use them on their own models or models of BioModels.

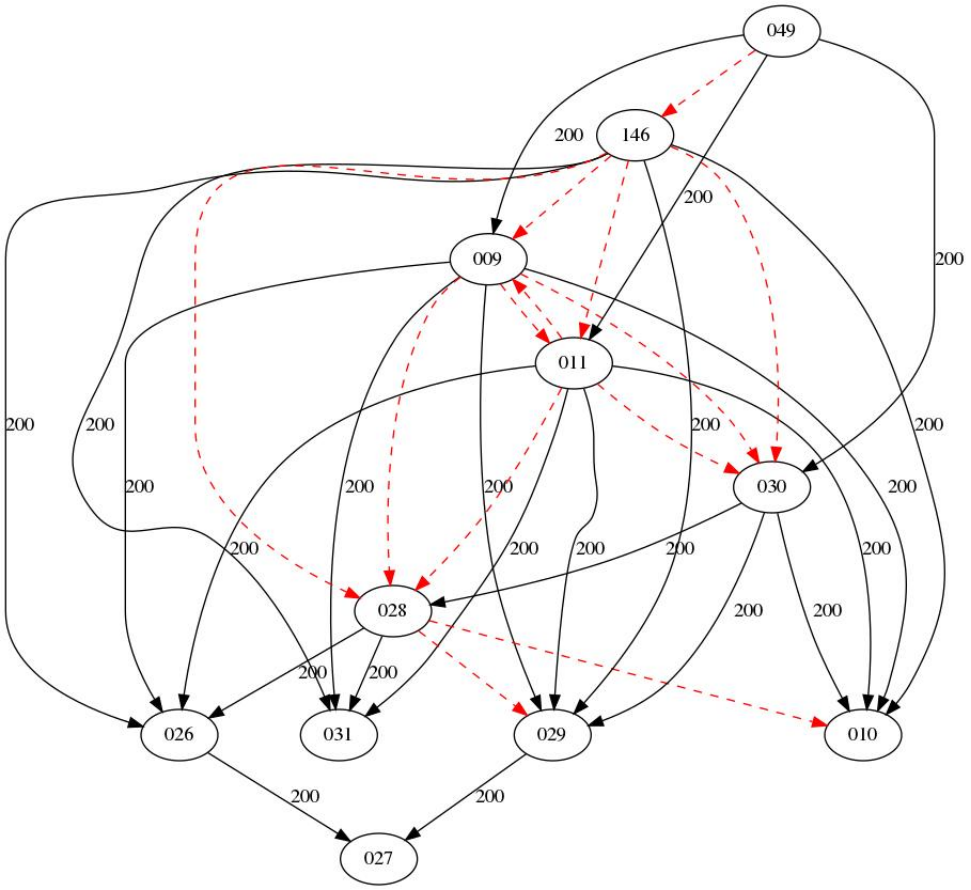

Figure 6-10.: SEPIs on MAPK without pattern reduction.

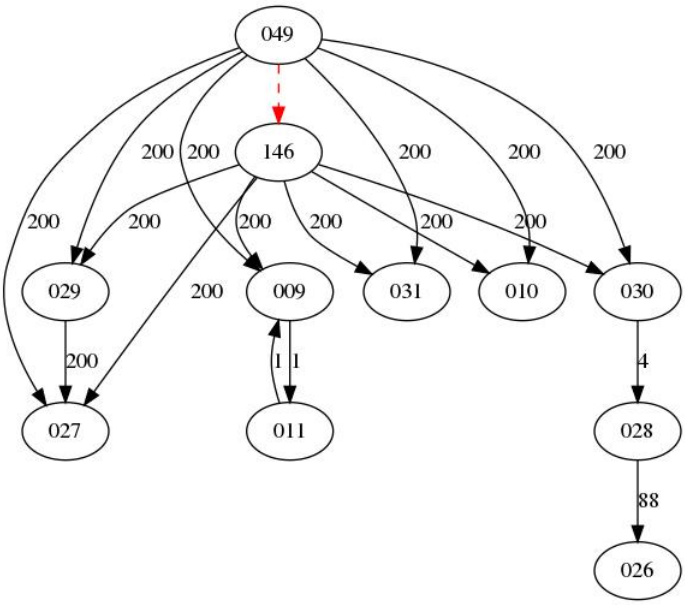

Figure 6-11.: SEPIs on MAPK with pattern reduction.

7. Bounds

7.1. Motivations

SEPI can be used to answer a yes/no question "is graph G' a reduced version of G?". However, since SEPI is not a total order, some pairs of graphs are incomparable. Instead of using a yes/no question, bounds (GLB or LUB) of G and G' can be computed. It corresponds to model an intersection or an union: to find a set of graphs G_c that are a common reduction of G and G' or that can be reduced both to G and G'.

7.2. Definitions

As a reminder $G = (V, A)$ and $G' = (V', A')$ are graphs, and o is an operation set among $\{m*\}$, $\{d\}$ and $\{m*, d\}$ with $m*$ the restricted merge operation.

Property 7.2.1 ($\rightarrow_o^*$ partial order) *The arrow $\rightarrow_o^*$ defines a partial order on the space of graphs.*

Proof: It is transitive, reflexive and anti-symmetric. □

Definition 7.2.1 (Set of lower bounds) *A set of lower bounds is defined by* $G \cap_o G' = \{H \mid G \rightarrow_o^* H \wedge G' \rightarrow_o^* H\}$

A *maximal* element of a set X is an element $x \in X$ such that $\forall y \in X, y \leq x$. x is the *maximum* of X if it is unique.

Definition 7.2.2 (Set of greatest lower bounds (GLB)) $\overline{G \cap_o G'}$ *is the set of $\rightarrow_o^*$-maximal elements of* $G \cap_o G'$.

Definition 7.2.3 (Set of upper bounds) *A set of upper bounds is defined by* $G \cup_o G' = \{H \mid H \rightarrow_o^* G \wedge H \rightarrow_o^* G'\}$

A *minimal* element of a set X is an element $x \in X$ such that $\forall y \in X, x \leq y$. x is the *minimum* of X if it is unique.

Definition 7.2.4 (Set of least upper bounds (LUB)) $\underline{G \cup_o G'}$ *is the set of $\rightarrow_o^*$-minimal elements of* $G \cup_o G'$.

Observation $G \cap_{m*d} G'$ is not empty.

If G and G' are connected and have at least an arc, $G \cap_{m*} G'$ is not empty.

The empty graph is in $G \cap_{m*d} G'$.

If G is connected and has at least an arc (s, r) or (r, s), with s a species vertex and r a reaction vertex, then by connectivity all its species vertices can be recursively merged with s and all its reaction vertices can be merged with r. The same goes for G'.

Observation $G \cup_{m^*d} G'$ is not empty.

It contains $G \cup_d G'$, which is not empty (it contains the disjoint union $G \uplus G'$).

However, for $G \cup_{m^*} G'$, the construction of $G \times G'$ can give disconnected graphs that are not compatible with the merge restriction.

Observation $G \cup_{m^*} G'$ can be empty.

G is defined as $G = (\{s_1, s_2\}, \{r\}, \{(s_1, r), (r, s_2)\})$ and G' as $G' = (\{s'\}, \{r'_1, r'_2\}, \{(r'_1, s'), (s', r'_2)\})$. Figure 7-1 represents both graphs.

Figure 7-1.: Graphs G and G' such that $G \cup_{m^*} G' = \emptyset$.

For a vertex u in a graph H, the set of outgoing neighbours of u is defined by $\mathcal{N}^\bullet(u) = \{v \in V \mid (u, v) \in A\}$, and the set of incoming neighbours by ${}^\bullet\mathcal{N}(u) = \{v \in V \mid (v, u) \in A\}$. $\mathcal{N}^\bullet(u) \neq \emptyset$ and ${}^\bullet\mathcal{N}(u) \neq \emptyset$ are invariant for any merge* operation. Hence, if H is a graph such that $H \to^*_{m^*} G$, it cannot have species vertex u with both $\mathcal{N}^\bullet(u) \neq \emptyset$ and ${}^\bullet\mathcal{N}(u) \neq \emptyset$. Moreover, H has to be connected, so there must be a reaction vertex t such that $\mathcal{N}^\bullet(t) \neq \emptyset$ and ${}^\bullet\mathcal{N}(t) \neq \emptyset$. This vertex cannot be sent to neither to r'_1 nor r'_2 in G', so H cannot verify $H \to^*_{m^*} G'$.

Additional properties of GLB and LUB can be found in appendix B.

Figure 7-2 gives examples of graph pairs such that $\overline{G \cap_o G'}$ and $\underline{G \cup_o G'}$ are not singletons.

Graphs G, G', H and H' show there is no unicity of GLB and LUB for the operation $o = m^*$. Relations $\to^*_{m^*}$ are indicated with dashed arrows, the fusions can be recognized with the names of H and H' vertices. In this example, $H, H' \in \overline{G \cap_{m^*} G'}$ and $G, G' \in \underline{H \cup_{m^*} H'}$.

Definition 7.2.5 (GLB decision problem) *The Greatest lower bounds problem is the decision problem:*

Instance: Two Graphs G, G'

Question: Is there a SEPI glb for G and G' of size

greater then k?

Theorem 7.2.1 *The GLB decision problem is NP-complete.*

Proof: SEPI decision problem "Is there a SEPI from G_1 to G_2" can be encoded as "Is there a SEPI glb of G_1 and G_2 of size $\mid G_2 \mid$". □

Definition 7.2.6 (LUB decision problem) *The Least upper bounds problem is the decision problem:*

Instance: Two Graphs G, G'

Question: Is there a SEPI lub for G and G' of size smaller then k?

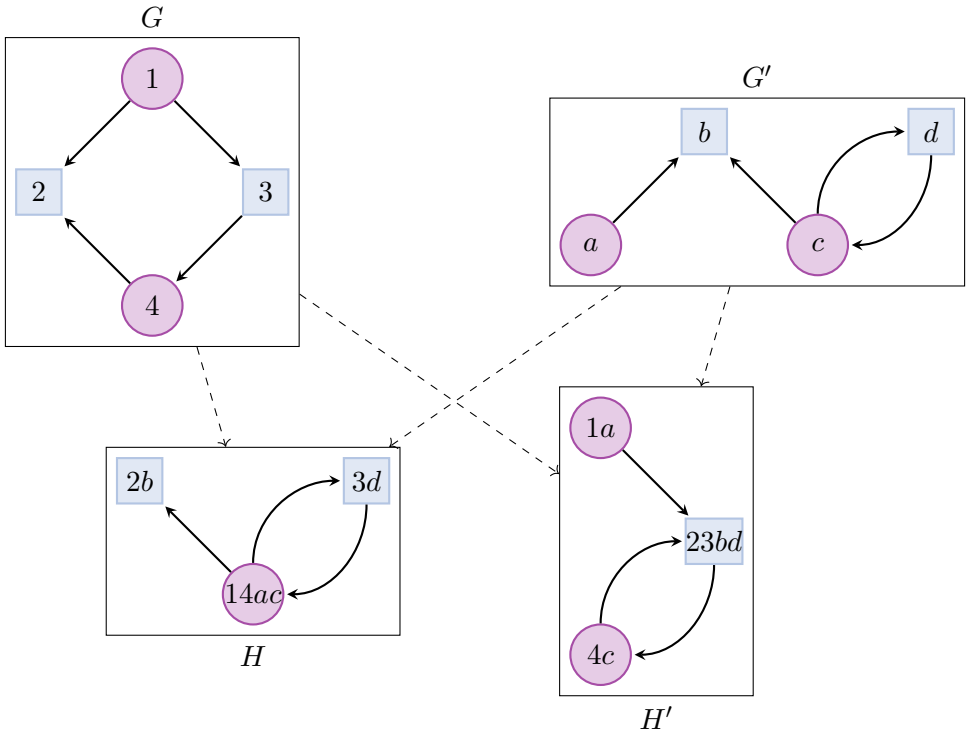

Figure 7-2.: Illustration no unicity.

Theorem 7.2.2 *[1] The LUB decision problem is NP-complete.*

7.3. Implementation

Implementing GLB problem or LUB problem is more difficult that the SEPI decision problem because the graph size is not known.

The SEPI GLB problem is the problem of finding a maximal G such that $G^1 \overset{SEPI}{\rightsquigarrow} G$ and $G^2 \overset{SEPI}{\rightsquigarrow} G$. Maximality should be with respect to the SEPI partial order, but as a proxy maximality will be with respect to the number of vertices. The encoding of SEPI will be changed to find two graph morphisms μ^1 from G^1 to G and μ^2 from G^2 to G. A MAX-SAT solver will be used to maximise the size of G which is bounded by $m' = min(\mid G^1 \mid, \mid G^2 \mid)$.

For the following implementation G_1 will be the smallest graph and G_2 the biggest graph. S' (respectively R') represents potential specie vertices (respectively reaction vertices) and is an ordered set $s'_0 = \bot < s'_1 < \ldots < s'_{p'}$ (respectively $r'_0 = \bot < r'_1 < \ldots < r'_{q'}$), with $p' = \mid S^1 \mid$ and $q' = \mid R^1 \mid$. s'_0 (respectively r'_0) is the image of deleted specie (respectively reaction) vertices.

Morphisms μ^1 and μ^2 define which vertices of S' and R' are kept as vertices and arcs of G.

As the coding is the same for μ^1 and μ^2, only the encoding of μ^i will be described.

Notations

- For $u \in S^i$ (resp. R^i) and $u' \in S'$ (resp. R'), $\mathbf{m}^i_{u,u'} = 1$ if and only if $\mu^i(u) = u'$,
- For $u \in S^i$ (resp. R^i) and $u' \in S'$ (resp. R'), $\mathbf{m}^{i,<}_{u,u'} = 1$ if and only if $\mu^i(u) < u'$.

  Left totality clauses are ensuring that each vertices of the starting graph have at least one image (or s'_0 or r'_0 for $\perp$) in the targeted graph.

  Functionality clauses are ensuring that each vertices of the starting graph have only one image through the morphism.

- For $v' \in S' \cup R'$, $\mathbf{e}_{v'} = 1$ if and only if v' is in the image of $\mu^i(V^i)$, which means that vertex v' exists in the glb graph G.

  Covering of existing vertices clauses put together the two morphisms μ_1 and μ_2.

  Bound maximisation clauses allow to maximize the size of the search graph G, with MAX-SAT.

- For $(u, v) \in A^i$ and $(u', v') \in S' \times R' \cup R' \times S'$, $\mathbf{m}^i_{(u,v),(u',v')} = 1$ if and only if $\mu^i(u) = u'$ and $\mu^i(v) = v'$.

  Graph morphism clauses ensure that an arc has an image through the morphism μ_i if both its vertices have an image through μ_i.

- For $(u', v') \in S' \times R' \cup R' \times S'$, $\mathbf{a}_{u',v'} = 1$ if and only if (u', v') is in the image of $\mu^i(A^i)$, which means that the arc (u', v') exists in the glb graph G.

  Label epimorphism clauses link together the arcs through the two morphisms μ_1 and μ_2.

- Existing set normalisation clauses force existing vertices to be the smallest in S' and R'.

  The species (resp. reaction) vertices of the smallest initial graph (supposed to be G_1) are put in a total order $s^1_1 < \ldots < s^1_p$ (resp. $r^1_1 < \ldots < r^1_q$).

  $\mathbf{B}^S_{v,v'} = 1$ if and only if $\forall u \leq v, \mu(u) \leq v'$.

  Symmetry breaking clauses force the tuple of the image vertices of the smallest initial graph $(\mu(v_1), \mu(v_2), \ldots, \mu(v_n))$ to be written in the smallest lexicographical order, with respect to all possible permutations of the image set. To do this, we use a precedence constraint (see [41] and [1]) with the bounding functions B^S (resp. B^R) such that $B^S : S \to S'$ is defined by $B(v) = \max\{\mu(u), u \leq v\}$, or equivalently by induction : $B(v_1) = \mu^1(v_1)$ and $B(v_{i+1}) = \max(B(v_i), \mu(v_{i+1}))$. The condition on the lexicographical minimality of $(\mu(v_1), \mu(v_2), \ldots, \mu(v_n))$ is equivalent to impose that $B(v_{i+1}) - B(v_i) \leq 1 \forall i \in [\![0, n-1]\!]$.

Clauses

I Left totality.

$\forall s^i \in S^i,$ $\quad$ $\forall r^i \in R^i,$

i $cl(\bigvee_{s' \in S'} \mathbf{m}^i_{s^i,s'})$. $\quad$ i $cl(\bigvee_{r' \in R'} \mathbf{m}^i_{r^i,r'})$.

II Functionality.

$\forall s^i \in S^i, \forall s'_j \in S'$,

i $cl(\mathbf{m}^i_{s^i,s'_j} \implies \mathbf{m}^{i,<}_{s^i,s'_{j+1}})$,

ii $cl(\mathbf{m}^{i,<}_{s^i,s'_j} \implies \mathbf{m}^{i,<}_{s^i,s'_{j+1}})$,

iii $cl(\mathbf{m}^{i,<}_{s^i,s'_j} \implies \neg\mathbf{m}^i_{s^i,s'_{j+1}})$.

$\forall r^i \in R^i, \forall r'_j \in R'$,

i $cl(\mathbf{m}^i_{r^i,r'_j} \implies \mathbf{m}^{i,<}_{r^i,r'_{j+1}})$,

ii $cl(\mathbf{m}^{i,<}_{r^i,r'_j} \implies \mathbf{m}^{i,<}_{r^i,r'_{j+1}})$,

iii $cl(\mathbf{m}^{i,<}_{r^i,r'_j} \implies \neg\mathbf{m}^i_{r^i,r'_{j+1}})$.

III Covering of existing vertices.

$\forall s' \in S' \setminus \{s'_0\}, \forall s^i \in S^i$,

i $cl(\mathbf{e}_{s'} \Rightarrow \bigvee_{s^i \in S_i} \mathbf{m}^i_{s^i,s'})$,

ii $cl(\mathbf{m}^i_{s^i,s'} \Rightarrow \mathbf{e}_{s'})$.

$\forall r' \in R' \setminus \{r'_0\}, \forall r^i \in R^i$,

i $cl(\mathbf{e}_{r'} \Rightarrow \bigvee_{r^i \in R_i} \mathbf{m}^i_{r^i,r'})$,

ii $cl(\mathbf{m}^i_{r^i,r'} \Rightarrow \mathbf{e}_{r'})$.

IV Graph morphism.

$\forall (u^i, v^i) \in A^i$, $\forall (u', v') \in V'^2$,

i $cl(\mathbf{m}^i_{(u^i,v^i),(u',v')} \implies \mathbf{m}^i_{u^i,u'})$,

ii $cl(\mathbf{m}^i_{(u^i,v^i),(u',v')} \implies \mathbf{m}^i_{v^i,v'})$,

iii $cl((\mathbf{m}^i_{u^i,u'} \wedge \mathbf{m}^i_{v^i,v'}) \implies \mathbf{m}^i_{(u^i,v^i),(u',v')})$.

V Label epimorphism.

i $\forall (u^i, v^i) \in A^i$, $\forall (u', v') \in V'^2$, $cl(\mathbf{m}^i_{(u^i,v^i),(u',v')} \Rightarrow \mathbf{a}_{u',v'})$.

ii $\forall (u', v') \in V'^2$, $cl(\mathbf{a}_{(u',v')} \Rightarrow \bigvee_{(u^i,v^i) \in A^i} \mathbf{m}^i_{(u^i,v^i),(u',v')})$.

VI Existing set normalisation.

$\forall s' \in S' \setminus \{s'_0, s'_1\}$,

i $cl(\mathbf{e}_{s'} \Rightarrow \mathbf{e}_{s'-1})$.

$\forall r' \in R' \setminus \{r'_0, r'_1\}$,

i $cl(\mathbf{e}_{r'} \Rightarrow \mathbf{e}_{r'-1})$.

VII Symmetry breaking.

$\forall (s_k, s'_j) \in S^1 \times S'$

i $cl(\mathbf{B}_{s_{k+1},s'_j} \Rightarrow \mathbf{B}_{s_k,s'_j})$,

ii $cl(\mathbf{B}_{s_k,s'_j} \Rightarrow \mathbf{m}^<_{s_k,s'_{j+1}})$,

iii $cl(\mathbf{B}_{s_k,s'_j} \Rightarrow \mathbf{B}_{s_k,s'_{j+1}})$,

iv $cl(\mathbf{B}_{s_k,s'_j} \Rightarrow \mathbf{B}_{s_{k+1},s'_{j+1}})$,

v $cl(\mathbf{B}_{s_k,s'_j} \wedge \mu^<_{s_{k+1},s'_j} \Rightarrow \mathbf{B}_{s_{k+1},s'_j})$.

$\forall (r_k, r'_j) \in R^1 \times R'$

i $cl(\mathbf{B}_{r_{k+1},r'_j} \Rightarrow \mathbf{B}_{r_k,r'_j})$,

ii $cl(\mathbf{B}_{r_k,r'_j} \Rightarrow \mathbf{m}^<_{r_k,r'_{j+1}})$,

iii $cl(\mathbf{B}_{r_k,r'_j} \Rightarrow \mathbf{B}_{r_k,r'_{j+1}})$,

iv $cl(\mathbf{B}_{r_k,r'_j} \Rightarrow \mathbf{B}_{r_{k+1},r'_{j+1}})$,

v $cl(\mathbf{B}_{r_k,r'_j} \wedge \mu^<_{r_{k+1},r'_j} \Rightarrow \mathbf{B}_{r_{k+1},r'_j})$.

VIII Bound maximisation.

i $\forall v' \in V'$, $cl(\mathbf{e}_{v'})$.

7.4. Conclusion

Greatest Lower Bound definition in honourable in a logical point of view. It could achieve the goal of finding a common reduced graph between two graph not comparable with the SEPI relation. But the implementation is very heavy and it didn't gave exploitable results. GLB decision problem is more complex than SEPI decision problem and even the SEPI framework was to complex to be computed on big models of BioModels database.

GLB implementation is not usable on big graphs. Moreover, implementing LUB is more complex than implementing GLB because, unlike for a lower bounds, no upper bound of a common graph in known. Thus, no implementation of LUB has been made.

8. Conclusion

Summary

The present diploma thesis is aimed to improve the SEPI framework. This SEPI framework is very powerful and already succeeded to compute reductions between biochemical reaction networks. Never less it has three main limits. First, establishing whether two models are linked through a SEPI is complex and computationally expensive. Second, the number of SEPIs found can be huge, making an analysis of SEPI sets between two given graphs very difficult for biologists. Finally, some existing SEPIs do not have a biological interpretation.

This diploma thesis led to three combined ways to improve the existing framework.

One way consisted to redefine the decision problem into an optimisation problem to select solutions by maximising or minimising the number of vertex deletions. This solution reduced the SEPI set's size between two given graphs and made reduction easier to read for biologists.

A second way was to determine, together with biologists, restrictions on the merge operations in order to filter irrelevant reductions. A strict restriction succeeded to delete some SEPIs without biological interpretations.

Lastly, a preprocessing step has been introduced, consisting of rewriting graphs according to subgraph isomorphism relations. The number of timeouts decreased drastically between models containing specific patterns.

The impact of these three combined implementations has been evaluated on models taken from BioModels' database. Results demonstrated that it contributed to make the SEPI framework more relevant, efficient and functional.

SEPI framework is a part of Biochemical Abstract Machine (BIOCHAM) which is a modelling environment for systems biology. All improvements implemented in this project have been integrated into Biocham and are now available for biologists.

Axes of enhancements

Models from BioModels are usually written in Systems Biology Markup Language (SBML). They come from peer reviewed scientific literature and are not always written according to the same conventions. An upstream standardisation work could improve a detection of SEPI reduction between models.

When only a system of ODEs describes a model, reaction graphs can be automatically inferred with Biocham from expressions of kinetic rates. In some cases, inferred graphs are unexpected, due to not well-chosen kinetic expressions. For example, inhibitors are inferred where there should not be. Thus enhancement of Biocham functions could contribute to better results with SEPI framework.

Finally, reaction graphs used for SEPI are making abstraction of each species' roles in reactions. Inhibition and enzymes are equally treated. Labelling edges accordingly could improve consistency.

Appendices

A. Additional figures

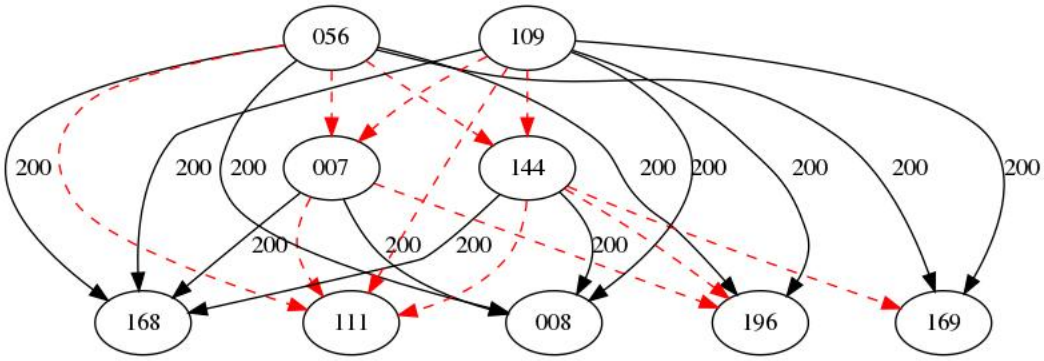

Figure A-1.: All SEPIs between models of Cell Cycle.

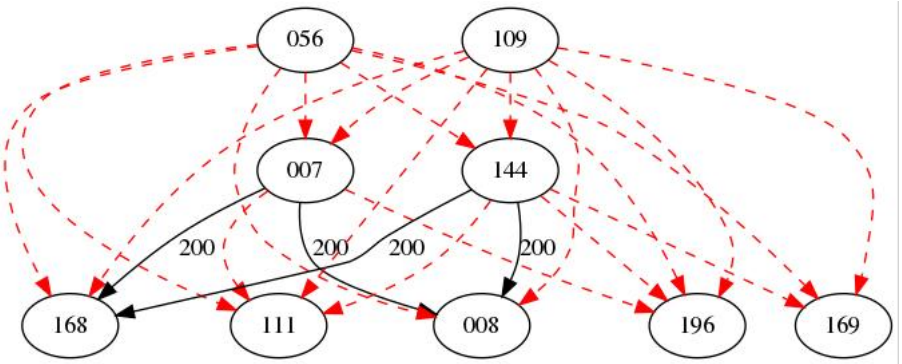

Figure A-2.: Cell Cycle - Minimisation of the number of deletions.

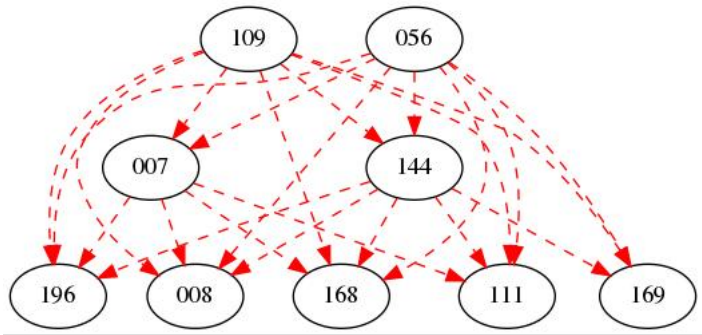

Figure A-3.: Cell Cycle - Maximisation of the number of deletions.

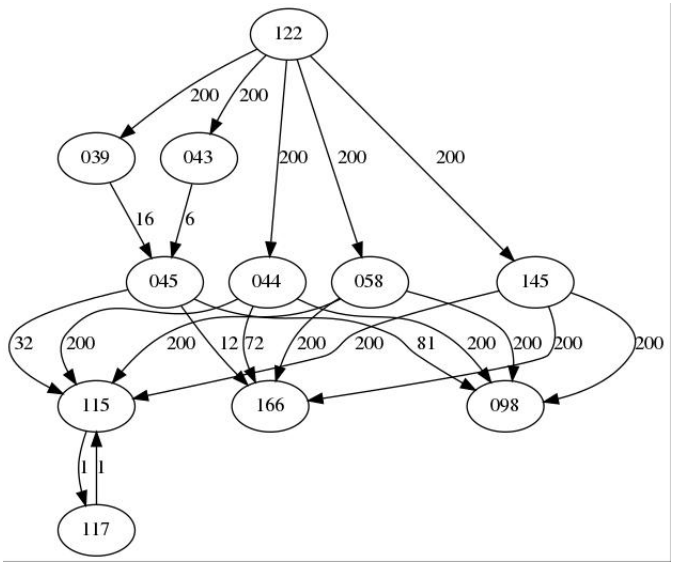

Figure A-4.: All SEPIs between models of Calcium Oscillations.

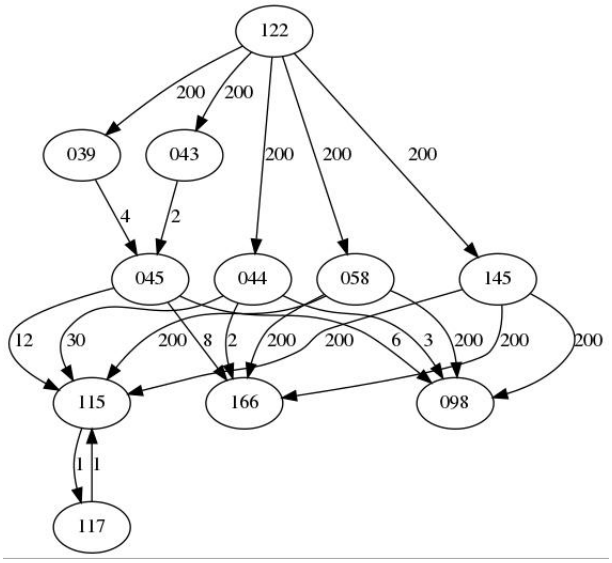

Figure A-5.: Calcium Oscillations - Minimisation of the number of deletions.

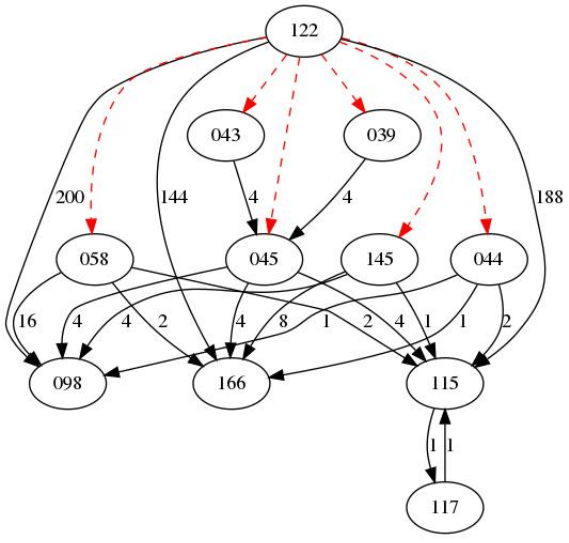

Figure A-6.: Calcium Oscillations - Maximisation of the number of deletions.

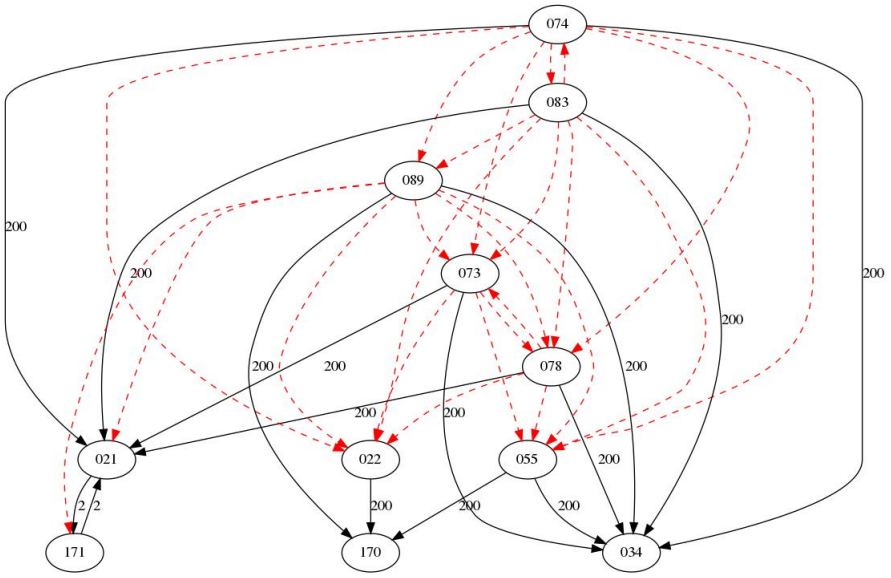

Figure A-7.: All SEPIs between models of Circadian Clock.

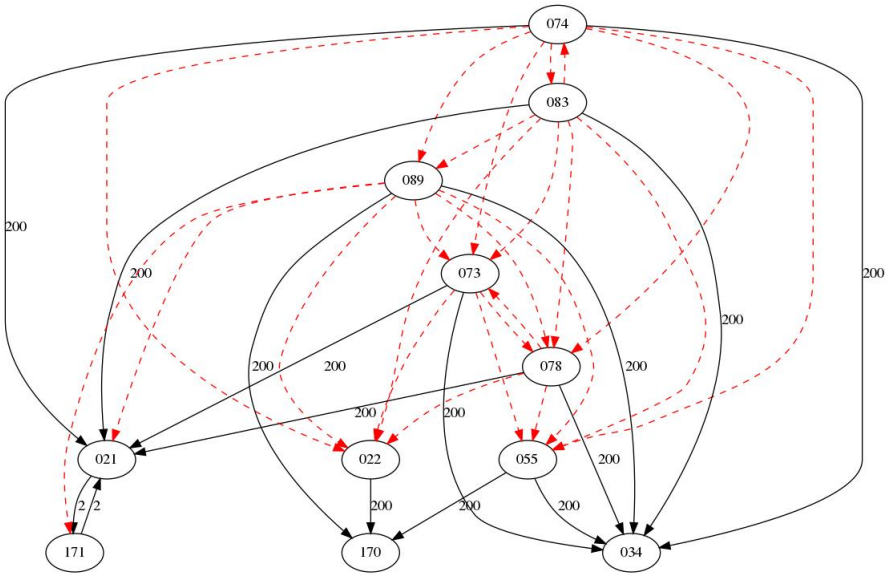

Figure A-8.: Circadian Clock - Minimisation of the number of deletions.

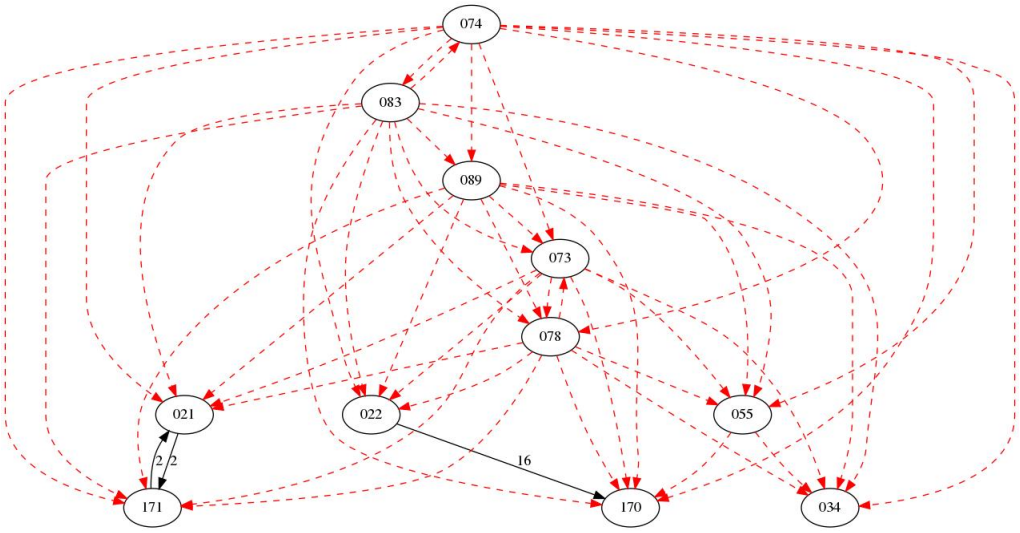

Figure A-9.: Circadian Clock - Maximisation of the number of deletions.

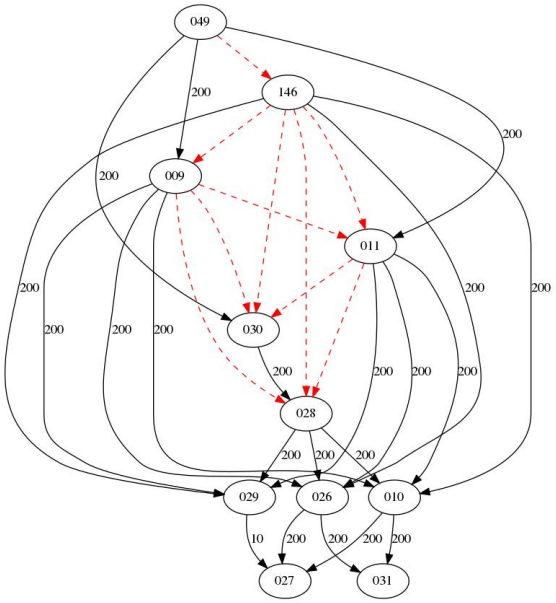

Figure A-10.: All SEPIs between models of MAPK.

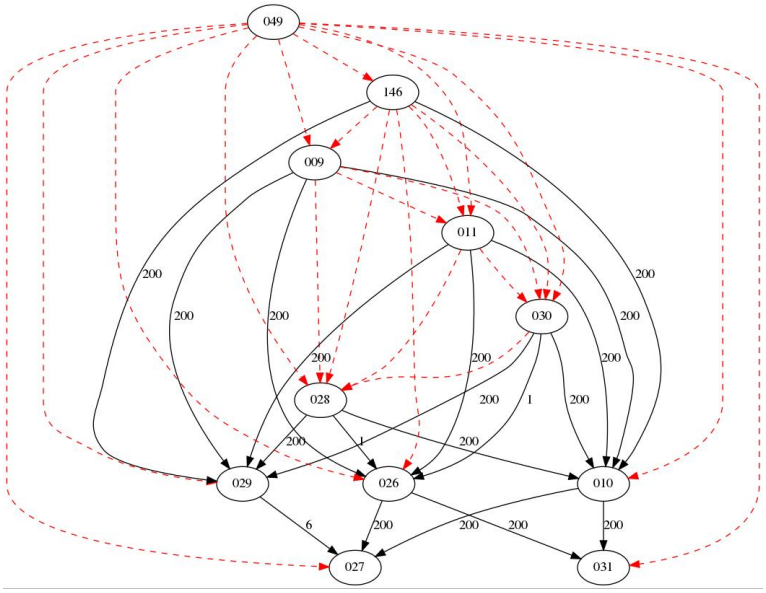

Figure A-11.: MAPK - Minimisation of the number of deletions.

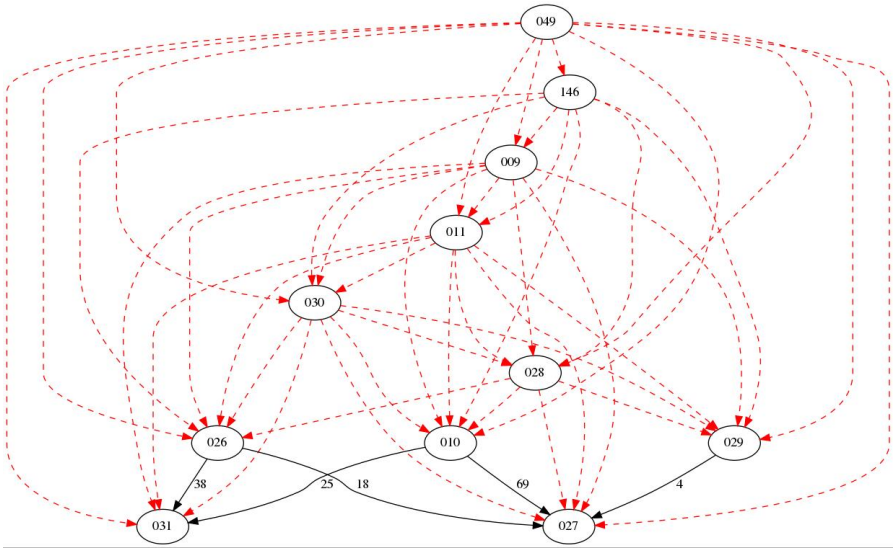

Figure A-12.: MAPK - Maximisation of the number of deletions.

B. Additional properties of GLB and LUB

These properties were studied by Eva Philippe.

Property B.0.1 *$\overline{G \cap_o G'}$ is finite.*

Proof: Delete and merge reduce the number of vertices, and graphs with less than $\min(|\ G\ |, |\ G'\ |)$ vertices are in finite number. □

Property B.0.2 *$\underline{G \cup_{m^*} G'}$ is not finite. It is unknown either $\underline{G \cup_{m^*d} G'}$ (i.e SEPI*) is finite or not.*

Proof: Figure B-1 and figure B-2 give an example of two graphs G and G' such that $\underline{G \cup_{m^*} G'}$ is infinite.

Figure B-3 illustrate how an infinity of graphs can be build. $\forall n \in \mathbb{N}\backslash\{0\}$, by concatenating n times a certain loop pattern, there is an infinity of graphs H_n in $\underline{G \cup_{m*} G'}$. A sequence of merge* operations that shows that $H_n \rightarrow^*_{m^*} G$ (respectively G') is given by folding the light-green parts by merging vertices a, a' and b, b' (respectively the light-blue parts), then merging the orange vertices, which have become 2-neighbours. Hence, $H_n \in G \cup_{m*} G'$.

In G (respectively G'), a pre-image of a blue/green part is necessarily a blue/green part in H_n. The only possible images for orange vertices in H_n are orange vertices in G and G'. There can be no merge* between blue and green parts because it would create a double arrow from an orange vertex. But there cannot be either a merge* inside a blue (respectively green) part, because then it could not be matched to G (respectively G'). Thus there can be no sequence of merge*-operations from H_n to H, also in $G \cup_{m*} G'$. □

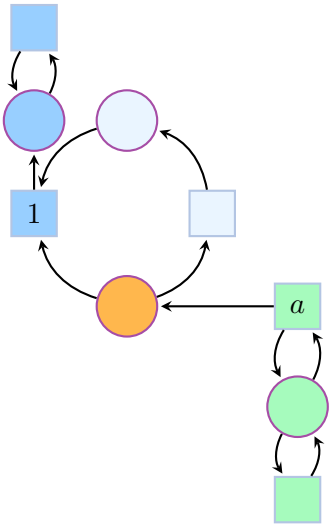

Figure B-1.: Graph *G*.

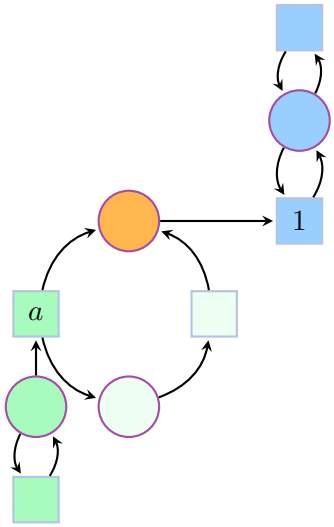

Figure B-2.: Graph *G'*.

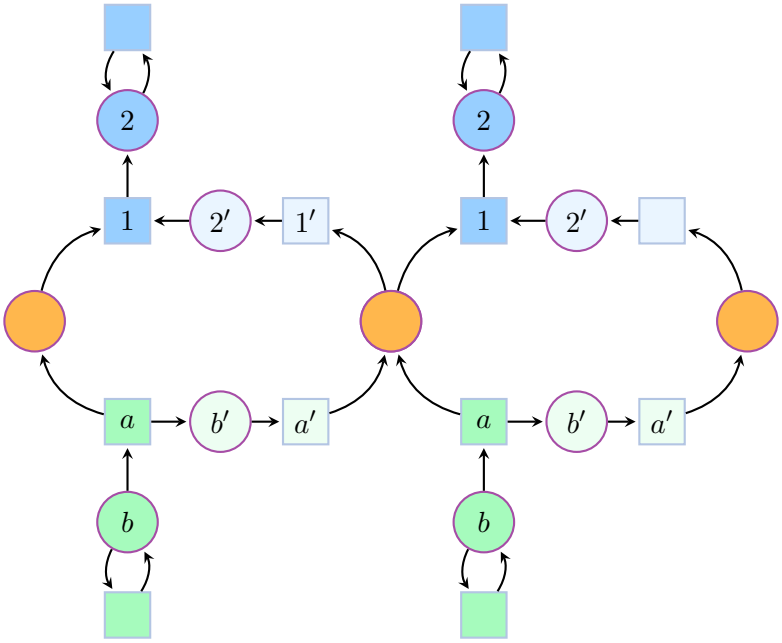

Figure B-3.: Infinite set illustration.

C. Implementation

This diploma thesis was aimed to improve a SEPI framework. This framework is a part of Biochemical Abstract Machine (BIOCHAM) which is a modelling environment for systems biology. It is developed by the Lifeware research team of Inria Saclay-Ile de France.

Biocham is a free software and is publicly available at `http://lifeware.inria.fr/biocham4`.

All improvements implemented in this project are integrated into Biocham and are available from version 4.2.6 at `gitlab.inria.fr/lifeware/biocham`. Documentation on how to execute the developed functions are in section 7.3. "Detecting model reductions" and section 7.4. "Pattern reduction" of the documentation created once Biocham is installed. Good-path merge restriction described in chapter 4 was not efficient enough to be integrated in the latest release. It can only be viewed in the branch *feature/sepi_refinement* of Biocham's repository. Bounds presented in chapter 7 are also in a separated branch called *feature/sepi_bounds*.

Models created for the evaluation and models curated from BioModels are available at `gitlab.inria.fr/evphilip/sepi-benchmark`.

If needed, sources of this report and sources of presentations given during internal seminars are available at `gitlab.inria.fr/obargain/diploma-thesis`.

List of Figures

List of Tables

Bibliography

[1] S. Gay, "Subgraph epimorphisms: Theory and application to model reductions in systems biology," Ph.D. dissertation, Université Paris Diderot, 2015.

[2] I. Tavassoly, J. Goldfarb, and R. Iyengar, "Systems biology primer: the basic methods and approaches," *Essays in biochemistry*, vol. 62, no. 4, pp. 487–500, 2018.

[3] V. Henri, *Lois générales de l'action des diastases.* Librairie Scientifique A. Hermann, 1903.

[4] N. Le Novere, B. Bornstein, A. Broicher, M. Courtot, M. Donizelli, H. Dharuri, L. Li, H. Sauro, M. Schilstra, B. Shapiro *et al.*, "Biomodels database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems," *Nucleic acids research*, vol. 34, no. suppl_1, pp. D689–D691, 2006.

[5] S. Sasagawa, Y.-i. Ozaki, K. Fujita, and S. Kuroda, "Prediction and validation of the distinct dynamics of transient and sustained erk activation," *Nature cell biology*, vol. 7, no. 4, p. 365, 2005.

[6] D. Shasha, J. T. Wang, and R. Giugno, "Algorithmics and applications of tree and graph searching," in *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems.* ACM, 2002, pp. 39–52.

[7] X. Yan, P. S. Yu, and J. Han, "Substructure similarity search in graph databases," in *Proceedings of the 2005 ACM SIGMOD international conference on Management of data.* ACM, 2005, pp. 766–777.

[8] N. Robertson and P. D. Seymour, "Graph minors. ii. algorithmic aspects of tree-width," *Journal of algorithms*, vol. 7, no. 3, pp. 309–322, 1986.

[9] L. Lovász, "Graph minor theory," *Bulletin of the American Mathematical Society*, vol. 43, no. 1, pp. 75–86, 2006.

[10] Y. Tian, R. C. Mceachin, C. Santos, D. J. States, and J. M. Patel, "Saga: a subgraph matching tool for biological graphs," *Bioinformatics*, vol. 23, no. 2, pp. 232–239, 2006.

[11] L. Cardelli, "Morphisms of reaction networks that couple structure to function," *BMC systems biology*, vol. 8, no. 1, p. 84, 2014.

[12] L. Noethen and S. Walcher, "Tikhonov's theorem and quasi-steady state," *Discrete Contin. Dyn. Syst. Ser. B*, vol. 16, no. 3, pp. 945–961, 2011.

[13] A. N. Tikhonov, "Systems of differential equations containing small parameters in the derivatives," *Matematicheskii sbornik*, vol. 73, no. 3, pp. 575–586, 1952.

[14] N. Fenichel, "Geometric singular perturbation theory for ordinary differential equations," *Journal of differential equations*, vol. 31, no. 1, pp. 53–98, 1979.

[15] S. Gay, S. Soliman, and F. Fages, "A graphical method for reducing and relating models in systems biology," *Bioinformatics*, vol. 26, no. 18, pp. i575–i581, 2010.

[16] S. Gay, F. Fages, T. Martinez, S. Soliman, and C. Solnon, "On the subgraph epimorphism problem," *Discrete Applied Mathematics*, vol. 162, pp. 214–228, 2014.

[17] R. Diestel, "Graph theory, volume 173 of," *Graduate texts in mathematics*, p. 7, 2012.

[18] J. Bang-Jensen and G. Z. Gutin, *Digraphs: theory, algorithms and applications*. Springer Science & Business Media, 2008.

[19] J. Van Leeuwen and J. Leeuwen, *Handbook of theoretical computer science*. Elsevier, 1990, vol. 1.

[20] L. Fuchs, *Partially ordered algebraic systems*. Courier Corporation, 2011, vol. 28.

[21] M. Davis and H. Putnam, "A computing procedure for quantification theory," *Journal of the ACM (JACM)*, vol. 7, no. 3, pp. 201–215, 1960.

[22] M. Davis, G. Logemann, and D. Loveland, "A machine program for theorem-proving," *Communications of the ACM*, vol. 5, no. 7, pp. 394–397, 1962.

[23] J. P. Marques-Silva and K. A. Sakallah, "Grasp: A search algorithm for propositional satisfiability," *IEEE Transactions on Computers*, vol. 48, no. 5, pp. 506–521, 1999.

[24] J. P. M. Silva and K. A. Sakallah, "Grasp: A new search algorithm for satisfiability," *Proceedings of the International Conference on Computer-Aided Design*, 1996.

[25] M. W. Moskewicz, C. F. Madigan, Y. Zhao, L. Zhang, and S. Malik, "Chaff: Engineering an efficient sat solver," in *Proceedings of the 38th annual Design Automation Conference*. ACM, 2001, pp. 530–535.

[26] M. Y. Vardi, "Boolean satisfiability: theory and engineering," *Communications of the ACM*, vol. 57, no. 3, pp. 5–5, 2014.

[27] D. Challenge, "Satisfiability: Suggested format," *DIMACS Challenge. DIMACS*, 1993.

[28] R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of computer computations*. Springer, 1972, pp. 85–103.

[29] M. Marhl, T. Haberichter, M. Brumen, and R. Heinrich, "Complex calcium oscillations and the role of mitochondria and cytosolic proteins," *Biosystems*, vol. 57, no. 2, pp. 75–86, 2000.

[30] J. M. Borghans, G. Dupont, and A. Goldbeter, "Complex intracellular calcium oscillations a theoretical exploration of possible mechanisms," *Biophysical chemistry*, vol. 66, no. 1, pp. 25–41, 1997.

[31] C.-Y. Huang and J. E. Ferrell, "Ultrasensitivity in the mitogen-activated protein kinase cascade," *PNAS*, vol. 93, no. 19, pp. 10 078–10 083, Sep. 1996.

[32] 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*, vol. 97, no. 11, pp. 5818–5823, May 2000.

[33] A. Ignatiev, A. Morgado, and J. Marques-Silva, "Rc2: a python-based maxsat solver," *MaxSAT Evaluation 2018*, p. 22.

[34] A. Zha, "Qmaxsat in maxsat evaluation 2018," *MaxSAT Evaluation 2018*, p. 21.

[35] S. Joshi, P. Kumar, V. Manquinho, R. Martins, A. Nadel, and S. Rao, "Open-wbo-inc in maxsat evaluation 2018," *MaxSAT Evaluation 2018*, p. 16.

[36] F. Bacchus, M. J. Jarvisalo, R. Martins *et al.*, "Maxsat evaluation 2018," 2018.

[37] R. M. Karp, "Reducibility among combinatorial problems," in *Proceedings of a Symposium on the Complexity of Computer Computations*, R. E. Miller and J. W. Thatcher, Eds., IBM Research Symposia Series. New York, NY: Plenum Press, 1972, pp. 85–103.

[38] N. I. Markevich, J. B. Hoek, and B. N. Kholodenko, "Signaling switches and bistability arising from multisite phosphorylation in protein kinase cascades," *Journal of Cell Biology*, vol. 164, no. 3, pp. 353–359, Feb. 2004.

[39] D. Moreland, W. Gentner, J. Hilton, and K. Hill, "Studies on the mechanism of herbicidal action of 2-chloro-4, 6-bis (ethylamino)-s-triazine." *Plant physiology*, vol. 34, no. 4, p. 432, 1959.

[40] N. E. Good, "Inhibitors of the hill reaction," *Plant physiology*, vol. 36, no. 6, p. 788, 1961.

[41] T. Walsh, "Symmetry breaking using value precedence," in *ECAI*, vol. 6, 2006, pp. 168–172.