Finding minimal P/T-invariants as a CSP

Sylvain Soliman

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

Abstract. We present here a way to compute the minimal semi-positive invariants of a Petri net representing a biological reaction system, as resolution of a CSP. The use of Petri-nets to manipulate those models and make available a variety of tools is quite old, and recently analyses based on invariant computation for biological models have become more and more frequent, especially in the context of module decomposition. In our case, this analysis brings both qualitative and quantitative information on the models, in the form of conservation laws, consistency checking, etc. thanks to finite domain constraint programming. It is noticeable that some of the most recent optimizations of standard invariant computation techniques in Petri-nets correspond to well-known techniques in CSPs, like symmetry-breaking. A simple prototype based on GNU-Prolog's FD solver, and including symmetry detection and breaking, was incorporated into the BIOCHAM modelling environment. Some illustrative examples and a few benchmarks are provided.

1 Introduction

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

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

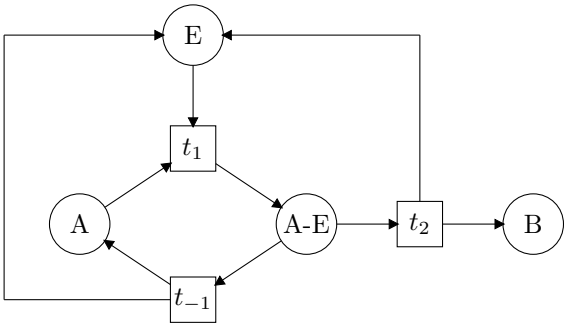
We present here a very simple way to incorporate invariant computation in an existing biological modelling tool, using constraint programming with symmetry detection and breaking. We compare it to other approaches and evaluate it, for the case of P-invariants, on some examples of various sizes, like the MAPK cascade models of [7] and [8]. This experimentation is done through an implementation of the described method in the BIOCHAM modelling environment[1] [9,10].

2 Petri-net view of a reaction model

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

The classical Petri-net view of a reaction model is simply to associate biochemical *species to places* and biochemical *reactions to transitions*.

Example 1. For instance the enzymatic reaction written (in BIOCHAM-like syntax), `A + E <=> A-E => B + E` corresponds to the following Petri-net :



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

P (resp. T) invariants are defined, as usual, as vectors V representing a multiset of places (resp. of transitions) such that $V \cdot I = 0$ (resp. $I \cdot V = 0$) where I is the *incidence matrix* of the Petri net, i.e. I_{ij} is the number of arcs

[1] At review time the version containing P-invariant computation might not have been released, but only in beta versions available at `http://www-rocq.inria.fr/~soliman/Biocham.dmg`

from transition i to place j, minus the number of arcs from place j to transition i. Intuitively, a P-invariant is a multiset representing a weighting of the places and such that any such weighted marking remains invariant by any firing; a T-invariant represents a multiset of firings that will leave invariant any marking (see also section 4). As explained in introduction, for reaction models these invariants are used for flux analysis, variable simplification through conservation law extraction, module decomposition, etc.

3 Related work

To compute the invariants of a Petri net, especially if this computation is combined with other Petri-net analyses, like sinks and sources, traps, deadlocks, etc. the most natural solution is to use a Petri-net dedicated tool like INA, PiNA, or Charlie for instance through the interface of Snoopy [11], which will soon allow the import of SBML models as Petri-nets. Standard integer methods like Fourier-Motzkin elimination will then provide an efficient means to compute P or T-invariants. These methods however generate lots of candidates which are afterwards eliminated and also need to incorporate some means (like equality class definition) to avoid combinatorial explosion at least in some simple cases, as explained in section 5.

Another way to extract the minimal semi-positive invariants of a model is to use one of the software tools that provide this computation for biological systems, generally as "conservation law" computation, and based on linear algebra methods like QR factorization [12]. This is the case for instance of the METATOOL [13] and COPASI [14] tools. The idea is to use a linear relaxation of the problem, which suits well very big graphs, but needs again *a posteriori* filtering of the candidate solutions. Moreover, these methods do not incorporate any means of symmetry elimination (see section 5).

4 Finding invariants as a Constraint Solving Problem

We will illustrate our new method for computing the invariants with the case of P-invariants (but T-invariants, being dual, would work in the same fashion). For a Petri net with p places and t transitions ($L_i \rightarrow R_i$), a P-invariant is a vector $V \in \mathbb{N}^p$ s.t. $V \cdot I = 0$, i.e. $\forall 1 \leq i \leq t\ V \cdot L_i = V \cdot R_i$. Since those vectors all live in $\mathbb{N}^p$, it is quite natural to see this as a CSP with t (linear) equality constraints on p Finite Domains variables.

Example 2. Using the Petri-net of example 1 we have:

```
A + E => A-E
  A-E => A + E
  A-E => B + E
```

This results in the following equations:

$$A + E = AE \tag{1}$$
$$AE = A + E \tag{2}$$
$$AE = B + E \tag{3}$$

where obviously equation (2) is redundant.

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

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

Now, to ensure minimality the labelling is invoked from small to big values and a branch and bound procedure is wrapped around it, maintaining a partial base $\mathcal{B}$ of P-invariant vectors and adding the constraint that a new vector V is solution if $\forall B \in \mathcal{B}\ \prod_{B_i \neq 0} V_i = 0$, which means that its support is not bigger than that of any vector of the base.

Unfortunately, even with the last constraint, no search heuristic was found that makes removing subsumed P-invariants unnecessary. Thus, if a new vector is added to $\mathcal{B}$, previously found vectors with a bigger support must be removed.

This algorithm was implemented directly into BIOCHAM [9], which is programmed in GNU-Prolog, and allowed for immediate testing.

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

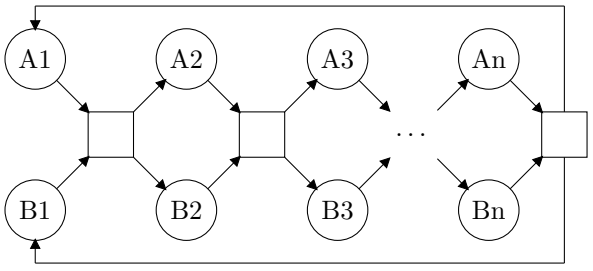
- $E = AE = 1$ and $A = B = 0$
- $A = B = AE = 1$ and $E = 0$

5 Equality classes

The problem of finding minimal semi-positive invariants is clearly EXPSPACE since there can be an exponential number of such invariants. For instance the model given in example 5 has 2^n minimal semi-positive P-invariants (each one with either A_i or B_i equal to 1 and the other equal to 0).

Example 5.

```
A1 + B1 => A2 + B2
A2 + B2 => A3 + B3
...
An + Bn => A1 + B1
```



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

This classical CSP optimization is enough to avoid most of the trivial exponential blow-ups and corresponds to the initial phase of *parallel places* detection and merging of the equality classes optimization for the standard Fourier-Motzkin algorithm [15]. Note however that in that method, classes of equivalent variables are detected and eliminated before and *during* the invariant computation, which would correspond to local symmetry detection and was not implemented in our prototype.

Moreover, in [15], *equality class* elimination is done through replacement of the symmetric places by a representative place. The full method reportedly improves by a factor two the computation speed. Even if in the context of the original article this is done only for ordinary Petri-nets (only one edge from one place to a transition and from one transition to one place), we can see that it can be even more efficient to use this replacement technique in our case:

Example 6.

```
...
A + B => 4*C
...
```

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

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

6 Example, the MAPK Cascade

The MAPK signal transduction cascade is a well studied system that appears in lots of organisms and is very important for regulating cell division [16]. It is composed of layers, each one activating the next, and in detailed models shows two intertwined pathways conveying EGF and NGF signals to the nucleus.

A simple MAPK cascade model, that of [17] without scaffold, is used here as an example to show the results of P-invariant computation.

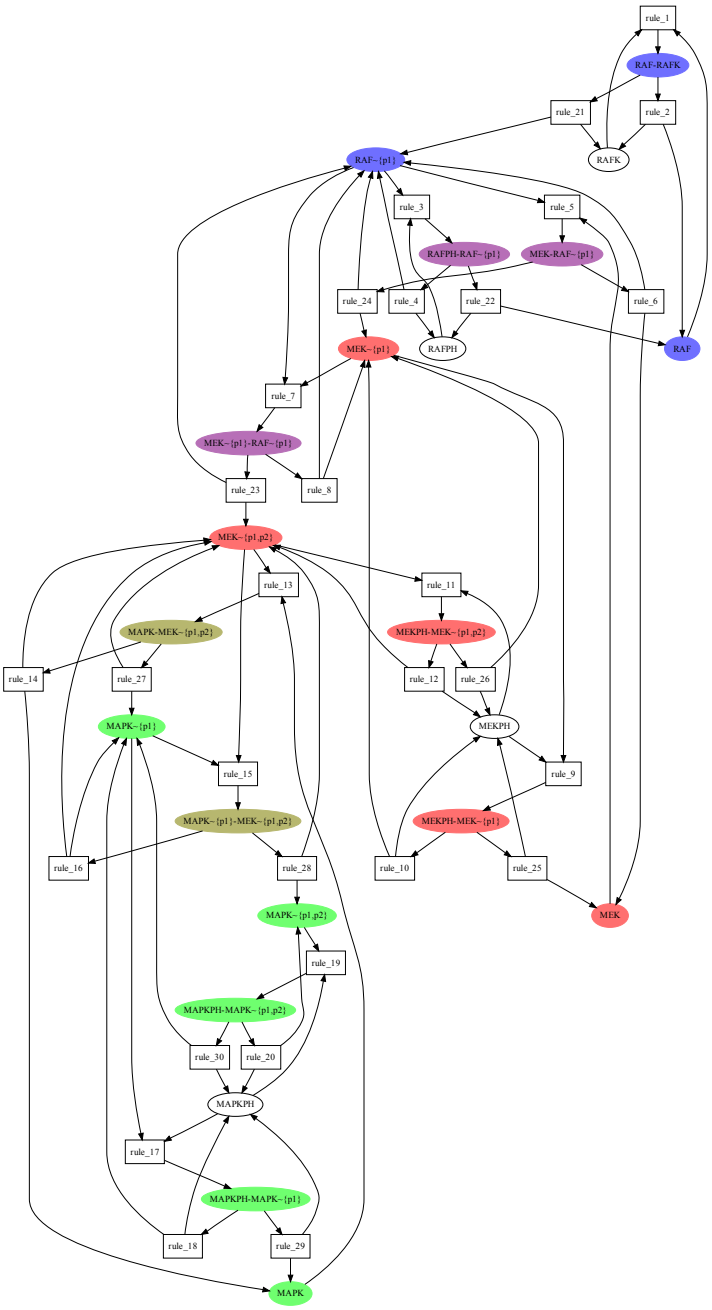


Fig. 1. 3 of the 7 P-invariants found in the MAPK cascade model of [17]. The blue one (RAF), the pink one (MEK) and the green one (MAPK) with intersections in purple (blue+pink) and khaki (pink+green).

Seven minimal semi-positive P-invariants are found almost instantly: RAFK, RAFPH, RAF, MEKPH, MEK, MAPKPH, MAPK. Three of them are depicted in figure 1, the full list is given in table 1.

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

Table 1. P-invariants of the MAPK cascade model of [17]

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

7 Evaluation on other examples

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

Model	transit.	places	P-invar.	time (s)	Invariant size
Schoeberl's MAPK [8]	125	105	14	<1	from 2 to 44
Curie's E2F/Rb [18]	∼500	∼400	79	∼10	from size 1 (EP300) to about 230 (E2F1 box)
Kohn's map [19]	∼800	∼500	65	∼40	from size 1 (Myt1) to about 200 (pRb or cdk2)

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

We could not compare our results with those provided in [12] since the models they use, coming from metabolic pathways flux analyses, do not have an integer stoichiometry matrix, however the examples of table 2 show the feasibility of P-invariant computation by constraint programming for quite big networks.

Note that for networks of this size, the upper bound of the domain of variables had to be set manually (to a reasonable value like 8 since actually only 2 or 3 was needed in all the biological models we have encountered up to now). Otherwise, the only over-approximation of the upper bound found was the product of the *l.c.m.* of stoichiometric coefficients of each reaction, which explodes really fast and leads to unnecessarily long computation. We thereby lose completeness, but it is not enforced either by QR-factorization methods, and does not seem to miss anything on real life examples.

8 Conclusion

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

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

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

References

1. Kanehisa, M., Goto, S.: KEGG: Kyoto encyclopedia of genes and genomes. Nucleic Acids Research **28** (2000) 27–30
2. Hucka, M., et al.: The systems biology markup language (SBML): A medium for representation and exchange of biochemical network models. Bioinformatics **19** (2003) 524–531
3. Reddy, V.N., Mavrovouniotis, M.L., Liebman, M.N.: Petri net representations in metabolic pathways. In Hunter, L., Searls, D.B., Shavlik, J.W., eds.: Proceedings of the 1st International Conference on Intelligent Systems for Molecular Biology (ISMB), AAAI Press (1993) 328–336
4. Gilbert, D., Heiner, M., Lehrack, S.: A unifying framework for modelling and analysing biochemical pathways using petri nets. In: CMSB'07: Proceedings of the fifth international conference on Computational Methods in Systems Biology. Volume 4695 of Lecture Notes in Computer Science., Springer-Verlag (2007)

5. Grafahrend-Belau, E., Schreiber, F., Heiner, M., Sackmann, A., Junker, B.H., Grunwald, S., Speer, A., Winder, K., Koch, I.: Modularization of biochemical networks based on classifica- tion of petri net t-invariants. BMC Bioinformatics **9** (2008)
6. Schuster, S., Fell, D.A., Dandekar, T.: A general definition of metabolic pathways useful for systematic organization and analysis of complex metabolic networks. Nature Biotechnology **18** (2002) 326–332
7. Chickarmane, V., Kholodenkob, B.N., Sauro, H.M.: Oscillatory dynamics arising from competitive inhibition and multisite phosphorylation. Journal of Theoretical Biology **244** (2007) 68–76
8. Schoeberl, B., Eichler-Jonsson, C., Gilles, E., Muller, G.: Computational modeling of the dynamics of the map kinase cascade activated by surface and internalized egf receptors. Nature Biotechnology **20** (2002) 370–375
9. Calzone, L., Fages, F., Soliman, S.: BIOCHAM: An environment for modeling biological systems and formalizing experimental knowledge. BioInformatics **22** (2006) 1805–1807
10. Fages, F., Soliman, S., Chabrier-Rivier, N.: Modelling and querying interaction networks in the biochemical abstract machine BIOCHAM. Journal of Biological Physics and Chemistry **4** (2004) 64–73
11. Heiner, M., Richter, R., Schwarick, M.: Snoopy - a tool to design and animate/simulate graph-based formalisms. In: Procedings of the International Workshop on Petri Nets Tools and APplications (PNTAP 2008), Marseille, ACM Digital Library (2008) to appear.
12. Vallabhajosyulaa, R.R., Chickarmane, V., Sauro, H.M.: Conservation analysis of large biochemical networks. BioInformatics (2005) Advance Access.
13. von Kamp, A., Schuster, S.: Metatool 5.0: fast and flexible elementary modes analysis. Bioinformatics **22** (2006) 1930–1931
14. Hoops, S., Sahle, S., Gauges, R., Lee, C., Pahle, J., Simus, N., Singhal, M., Xu, L., Mendes, P., Kummer, U.: Copasi – a complex pathway simulator. BioInformatics **22** (2006) 3067–3074
15. Law, C.F., Gwee, B.H., Chang, J.: Fast and memory-efficient invariant computation of ordinary petri nets. IEE Proceedings: Computers and Digital Techniques **1** (2007) 612–624
16. Roovers, K., Assoian, R.K.: Integrating the MAP kinase signal into the G1 phase cell cycle machinery. BioEssays **22** (2000) 818–826
17. Levchenko, A., Bruck, J., Sternberg, P.W.: Scaffold proteins may biphasically affect the levels of mitogen-activated protein kinase signaling and reduce its threshold properties. PNAS **97** (2000) 5818–5823
18. Calzone, L., Gelay, A., Zinovyev, A., Radvanyi, F., Barillot, E.: A comprehensive imodular map of molecular interactions in RB/E2F pathway. Molecular Systems Biology **4** (2008)
19. Kohn, K.W.: Molecular interaction map of the mammalian cell cycle control and DNA repair systems. Molecular Biology of the Cell **10** (1999) 2703–2734