Optimal Simulation and Maximally Concurrent Evolution - My Early Papers With Maciej Koutny

Ryszard Janicki

Department of Computing and Software, McMaster University,
Hamilton, Ontario, L8S 4K1 Canada
janicki@mcmaster.ca

1 Beginnings

I have met Maciej somehow around 1979/80, when he took my undergraduate course 'Introduction to Switching and Automata Theory'. The course was based on then popular textbook by M. A. Harrison, with the same title, which was also known for having plenty of very difficult problems as exercises. In the middle of term, I gave one of these problems to the students as optional homework. The next class was almost a week later and only Maciej said that he had a solution. His solution was very clever, much more elegant and shorter than the one I had, and he presented it in such nonchalant way, as he did in less that twenty minutes just before the class. I was very impressed as it took me some substantial time to produce my inferior and longer solution. Only about fifteen years later, when we were friends and research collaborators, he admitted that during this almost a week between classes, practically the only thing he thought of, was to try to solve this problem. But his presentation suggested otherwise.

Both him and, then his girlfriend, Marta Pietkiewicz, become interested in my research, Marta did Master Thesis under my supervision, Maciej did his Master Thesis under supervision of Antoni Mazurkiewicz and later PhD under my supervision.

Nevertheless I do not have any common paper with Maciej related to his PhD thesis. In fact I have no common paper related to their theses with any of my former Polish PhD and Masters students. I also have no common paper related to my PhD thesis with Antoni Mazurkiewicz, my PhD supervisor. In seventies and at least first half of eighties last century having papers with students, for most mathematicians and theoretical computer scientists, were considered a breaking of some well established code of honour. The code of honour of Polish School of Mathematics was still very observed, and most of people from theoretical computer science graduated mathematics. According to this code of honour, having papers with students (with rare well justified exceptions) was considered dishonourable. Sounds weird today but this was true then.

It was only in Canada, where to my great surprise, I have learned that having common papers with students is not only a virtue but almost a necessity for a successful grant application or promotion process. Since swimming against the current is seldom successful in long term, right now I have plenty papers with my graduate students, but our first common paper with Maciej was written few years after his PhD, and it dealt with the problem of maximal concurrency [13].

2 Maximal Concurrency and Optimal Simulations/Executions

Since 1986 I have published with Maciej a few dozens of papers on different subjects, the most known are our results on *relational structure semantics of concurrent systems*, which started with [7] and the most representative papers on this subject are still probably [11] and [12]. Nevertheless our first period of collaboration, roughly from 1985 to 1991 was devoted to maximally concurrent and optimal executions (simulations) of concurrent systems under step sequence observational semantics paradigm [4–6, 8, 9, 13], and additional assumption that concurrent behaviours are entirely specified by partial orders.

Among various semantics of executions in non-sequential systems, we can distinguish three widely accepted approaches. The first one, standard in Petri net based models [17], COSY paths expressions [10, 14], Asynchronous Automata [21] and many others [11], may intuitively be expressed as: '*execute as possible*'. This encompasses the whole range of possibly concurrent evolutions; from the sequential ones to the maximally concurrent ones through all the intermediate case, any possible execution is allowed. For example, assuming that observational semantics is defined in terms of step sequences, if the events a,b,c are independent, the following step sequences are considered as equivalent: $\{a,b,c\}$, $\{a,b\}\{c\}$, $\{a,c\}\{b\}$, $\{b,c\}\{a\}$, $\{a\}\{b,c\}$, $\{b\}\{a,c\}$, $\{c\}\{a,b\}$, $\{a\}\{b\}\{c\}$, $\{b\}\{a\}\{c\}$, $\{a\}\{c\}\{b\}$, $\{c\}\{a\}\{b\}$, $\{b\}\{c\}\{a\}$ and $\{c\}\{b\}\{a\}$.

The second one can be called '*execute as possible, but in sequence*', is called *interleaving semantics* and is widely used in all kinds of Process Algebras [1]. In this case for three independent events a,b,c, we have sequences: *abc*, *bac*, *acb*, *cab*, *bca*, *cba*.

The third one is usually expressed as '*execute as much as possible in parallel*' or '*execute as quick as possible*'. In this case we require that processes should not be lazy, and at each step of computation, the set of events executed must be a maximal non-conflict set. This greedy semantics is often used for some kind of Timed Petri Nets [20], some temporal logic applications [3], it is often easier to implement and has many nice algorithmic properties [18]. For our three independent events a,b,c, it will be represented by just one step $\{a,b,c\}$.

Our first common paper [13] (with Peter Lauer and Raymond Devillers as the remaining co-authors) was a successful attempt to answer the question: "*When is the semantics 'execute as much as possible in parallel' equivalent to the semantics 'execute as possible'?*". Various necessary and sufficient conditions for this property were provided and proved. We used COSY Path Expressions (first proposed by Peter Lauer and R. H. Campbell in [14], fully presented in [10]) to represent concurrent systems, and Vector Firing Sequences (proposed by Mike Shields in [19]) to represent concurrent behaviours, however the results can easily be translated into Elementary Petri Nets [16] and Mazurkiewicz Traces [15] extended to step sequences (cf. [4, 8]).

Employing maximal concurrency is an attractive idea, both conceptually and from the point of view of implementation. Unfortunately, as pointed out in [13], there are cases in which the greedy maximally concurrent execution is not sufficiently expressive. To show this case we take the net from Figure 1. The maximally concurrent execution can find only one deadlocked marking of the net, by following the step sequence $\{a,b\}\{d\}$. The other deadlocked marking, which might be reached by following the step sequence $\{b\}\{c\}$, is left undetected.

The process of verification of dynamic properties of non-sequential systems often involves some kind of reasoning about the complete state-space of the system, e.g. proving deadlock-freeness requires showing that it is not possible to reach a state in which no transition is enabled. Reasoning about the complete state-space of non-sequential systems has one very serious drawback, which is a combinatorial explosion of state-space. Even a simple concurrent system can generate many hundreds or thousands of states. To cope with this problem a number of sophisticated techniques have been developed, for example induction, reduced stat-space analysis, etc.

In [4] and [5] a possibility of defining a fully expressive reachability relation on the system's executions which would be a 'small' subset of the complete reachability relation. Such reduced reachability relation was called the *optimal simulation* (it was later renamed to *optimal execution* in [10]). Conceptually, the optimal simulation can be considered as a generalized version of maximal concurrency and, in some, but not so infrequent, cases it is just the maximal concurrency [4, 5]. The concept of *canonical step sequence*, a simple extension of canonical sequence for Mazurkiewicz traces [2, 13] plays crucial role. Figure 1 shows both the full reachability graph of a simple elementary net [16] and the reachability graph of the optimal simulation. The latter is smaller, but because in this case only two transitions *a* and *b* can be fired concurrently, the difference in size is not significant. However, in the case of net in Figure 2, the reachability graph of the optimal simulation is isomorphic to that in Figure 1, while (as one may easily check) the full reachability graph would hardly fit on a single page.

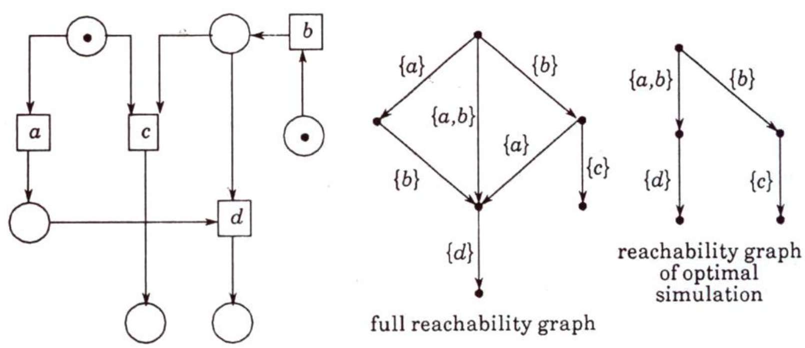

Fig. 1: Full reachability graph versus reachability graph of optimal simulation. Incompleteness of maximally concurrent semantics - the right deadlocked marking, after the step sequence $\{b\}\{c\}$ is not detected.

Optimal simulation enables reasoning about a number of dynamic properties of concurrent systems, and at the same time requires some minimal computational effort. In [4, 5], the optimal simulation had been defined in a very general step sequence setting which made it applicable to different models of concurrency, such as Petri nets, various process algebras or automata-based models.

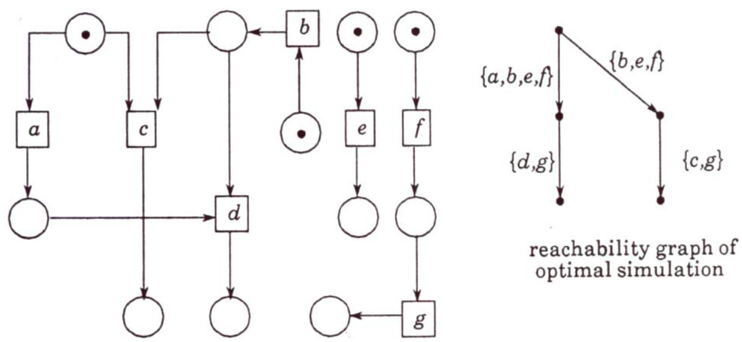

Fig. 2: A net with small reachability graph of optimal simulation and huge (not shown) full reachability graph.

The reachability graphs of finite state systema ca be regarded as *finite representations of reachability relations*. Since the optimal simulation provides the same information about relevant dynamic properties of the system as the full reachability relation [4, 5], the reachability graph of optimal simulation (i.e. it finite representation) and the full reachability graph may be considered equivalent. The optimal simulation is always a subset of full reachability, but of course it does not mean that the reachability graph of optimal simulation is always much smaller than the full reachability graph. However, in the case of concurrent systems exhibiting high degree of concurrency (i.e. those with many sequential components), the reachability graph of optimal simulation is much smaller that the full reachability graph. Thus it is advantageous to use optimal simulation as a tool to reduce the size of reachability graphs of concurrent systems.

Unfortunately, as opposed to the full reachability graph, in the general case it is not clear how to generate the reachability graph for optimal simulation. In some sense this is a negative side-effect of the generality of optimal simulation.

In [6] and [8], we have shown how the reachability graph for optimal simulation can be constructed for Elementary Petri Nets that can be decomposed onto finite state machines, and in [9] for concurrent systems that can be represented by asynchronous automata of [21].

The approach presented in [4–6, 8, 9, 13] was based on the assumption that concurrent behaviours (histories) are modeled by casual partial orders. We represented those partial orders by certain equivalence classes of step sequences, just generalizing the notion of Mazurkiewicz traces [15]. The relational structures and 'not later than' phenomenon, introduced in [7] and first time discussed in detail in [11, 12] are not considered in these papers.

The major methodological difference between our approach based on optimal simulation to minimize reachability graphs and virtually almost all others is that we do not try to minimize (or even deal with) the full reachability graph. All what we were trying to do was to buld a reachability graph which represents optimal simulation relation (a subset of full reachability). We then made a claim, based on the general properties of optimal simulation, that in majority of cases, such a graph is much smaller than the original full reachability graph.

I believe the concept of optimal simulation (or execution) and its usefulness was and still is a little bit unappreciated. The main reason could be that we never built any software that supported optimal simulation and never applied the ideas of optimal simulation to real concurrent systems as for example security protocols. We were than both relatively young recherches and neither of us had sufficient knowledge, experience and connections for successful huge grant application, that was needed for producing good experimental software.

I also believe that the problem of using optimal simulation/execution is still not closed and still has many potential applications. It may be worth to go back to it in some future time.

References

1. J. A. Begstra et al (eds.), *The Handbook of Process Algebras*, Elsevier Science 2000.
2. Cartier, P., Foata, D.: Problemes combinatoires de communication st rearrangements, *Lecture Notes in Mathematics* 85, Springer 1969.
3. Enjalbert, P., Michel, M.: Many-sorted temporal logic for multiprocesses systems, *Lecture Notes in Computer Science* 176, Springer 1984, 273-281.
4. Janicki, R., Koutny, M.: On Equivalent Execution Semantics of Concurrent Systems, *Lecture Notes in Computer Science* 266, Springer 1987, 89–103.
5. Janicki, R., Koutny, M.: Towards a Theory of Simulation for Verification of Concurrent Systems, Proc. of Parallel Architectures and Languages Europe '89, *Lecture Notes in Computer Science* 366, Springer 1989, 73-88.
6. Janicki, R., Koutny, M.: Net Implementation of Optimal Simulation, Proc. of Application and Theory of Petri Nets '90, Paris, France, June 1990.
7. Janicki, R., Koutny, M.: Invariants and Paradigms of Concurrency Theory. *Lecture Notes in Computer Science* 506, Springer 1991, 59–74.
8. Janicki, R., Koutny, M.: Using Optimal Simulations to Reduce Readability Graphs, *Lecture Notes in Computer Science* 531, Springer 1991, 166-175
9. Janicki, R., Koutny, M.: On Some Implementation of Optimal Simulation, *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 3 (1991), 231-250
10. Janicki, R., Lauer, P. E.: *Specification and Analysis of Concurrent Systems*. The COSY Approach, Springer 1992.
11. Janicki, R., Koutny, M.: Structure of Concurrency. *Theoretical Computer Science* 112, 5–52 (1993)
12. Janicki, R., Koutny, M.: Semantics of Inhibitor Nets. *Information and Computation* 123(1), 1–16 (1995)
13. Janicki, R., Lauer, P. E., Koutny, M., and Devillers, R.: Concurrent and Maximally Concurrent Evolution of Non-Sequential Systems, *Theoretical Computer Science*, 43, 213–238 (1986).
14. Lauer, P. E., Campbell, R. H.: Formal Semantics of a Class of High level Primitives for Coordinating Concurrent Processes, *Acta Informatica* 5, 297–332 (1975)
15. Mazurkiewicz, A.: Concurrent Program Schemes and Their Interpretation. TR DAIMI PB-78, Comp. Science Depart., Aarhus University (1977)
16. Nielsen, M., Rozenberg, G., Thiagarajan, P. S.: Behavioural Notions for Elementary Net Systems. Distributed Computing 4, 45–57 (1990)

17. Reisig, W.: *Petri Nets*, Springer 1985.
18. Salwicki, A., Müldner, T., On Algorithmic Properties of Concurrent Programs, *Lecture Notes in Computer Science* 125, Springer 1981, 169–197.
19. Shields, M. W.: Adequate Paths Expressions, *Lecture Notes in Computer Science* 70, Springer 1979, 249–265
20. Wang, J.: *Timed Petri Nets*, Kluwer Academic 1998
21. Zielonka, W.: Notes on Finite Asynchronous Automata, *RAIRO Inform. Théor. Appl.* 21:99–135, 1987.