

Convex Petri Nets

Nicola Cotumaccio¹, Catia Trubiani²

¹University of Helsinki, Yliopistonkatu 4, 00100 Helsinki, Finland

²Gran Sasso Science Institute, Viale Luigi Rendina, 26-28, 67100 L'Aquila, Italy

Abstract

Both Petri Nets and finite automata are prone to the state explosion problem. A recent line of research shows the state explosion in finite automata is captured by a notion of convexity. We outline how the result on finite automata can be exploited to suggest a new approach for studying Petri nets.

Keywords

Petri Nets, Convexity, Data Compression, Automata Theory

Petri nets are widely used to model distributed systems. The expressive power of this formalism comes to a price: Petri nets are prone to the state explosion problem [1]. A Petri net can describe a system that may have infinitely many states, thus making it difficult to analyze and simulate the system's behavior. Given a Petri net and an initial marking, the *reachability graph* is a (possibly infinite) graph describing all markings of the Petri nets reachable from the initial marking by firing transitions (see Figure 1). A classical result shows that it is possible to decide whether the reachability graph is finite by constructing the *Karp-Miller tree* of the Petri net. To limit the complexity of the reachability graph, it is possible to add (topological) constraints to the Petri nets that ensure that the reachability graph is small or at least finite, while still allowing the modeling of a wide range of systems (see [2] for a survey). A Petri net is *structurally bounded* if the reachability graph is finite for every choice of the initial marking. An important special case of structurally bounded Petri nets is that of *conservative* Petri nets, in which after firing any transition, the total number of tokens is the same as before firing the transitions. Alternatively, we can add constraints that depend on a parameter k to parameterize the size of the reachability graph as a function of k . For example, a Petri net is *k-bounded* if every marking reachable from the initial marking has the property that no place contains more than k tokens. In particular, an 1-bounded Petri net is *safe*.

Petri nets can be seen as an extension of finite automata. In automata theory, a similar state explosion occurs when we apply the *powerset construction* to convert a nondeterministic automaton (NFA) into a deterministic automaton (DFA) recognizing the same regular language: if an NFA has n states, then the equivalent DFA can have up to $2^n - 1$ states (see Figure 1). While the expressive power of NFAs is the same as the expressive power of DFAs, determinism is extremely helpful in studying the properties of an automaton. Some basic problems in automata theory – such as the *emptiness problem* and the *equivalence problem* – are PSPACE complete on NFAs but can be solved in almost-linear time on DFAs. The natural question is whether for automata it is possible to follow the same path that we sketched for Petri nets – adding constraints to the topology of the considered NFAs to control the state explosion.

A recent line of research in automata theory has shown that, indeed, a notion of *convexity* captures the complexity of the powerset construction and establishes an unexpected connection between data compression and automata theory, thus leading to *Wheeler automata* [3]. A Wheeler NFA is endowed with a total order on the set of all states. In Figure 2, the total order is given by the numbering of the states. First, the total order respects equally-labeled edges: for example, if we consider edges $(10, 5, b)$ and $(11, 6, b)$, we have $10 < 11$, and consistently we have $5 < 6$. Second, if we consider edges with distinct labels – such as $(5, 2, a)$ and $(11, 7, b)$ – we have $a < b$ and consistently the end states satisfy

PNSE'24, International Workshop on Petri Nets and Software Engineering, 2024

✉ nicola.cotumaccio@helsinki.fi (N. Cotumaccio); catia.trubiani@gssi.it (C. Trubiani)

🆔 0000-0002-1402-5298 (N. Cotumaccio); 0000-0002-7675-6942 (C. Trubiani)



© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

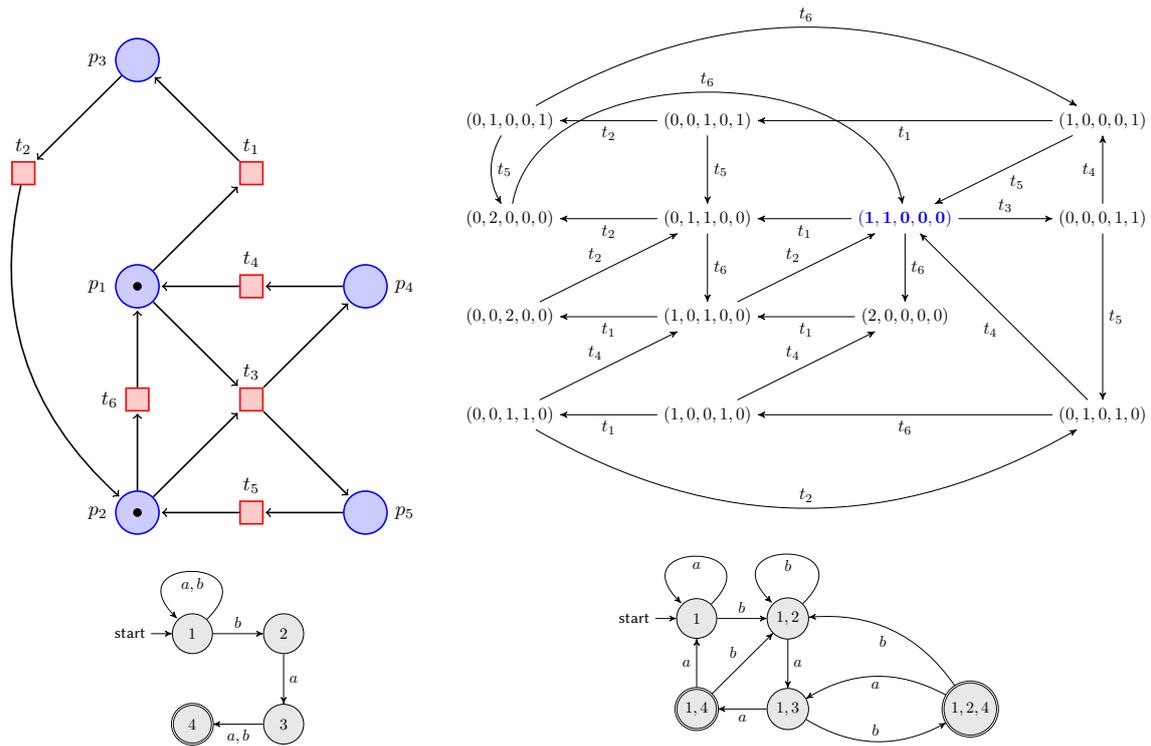


Figure 1: *Top left:* A (conservative) Petri Net, with initial marking $(1, 1, 0, 0, 0)$, meaning that p_1 and p_2 contain 1 token and p_3, p_4 and p_5 contain 0 tokens. *Top Right:* The corresponding reachability graph. *Bottom left:* An NFA. *Bottom right:* The equivalent DFA obtained by the powerset construction. Every state in the equivalent DFA corresponds to a nonempty set of states in the original NFA.

$2 < 7$. Wheeler automata admit a *compressed* representation that supports *efficient* pattern-matching queries. This is crucial in bioinformatics, where both time-efficient and space-efficient algorithms are required. For example, *de Bruijn automata* are used in genome assembly [4], and every de Bruijn automaton is a Wheeler automaton.

Surprisingly, if we start from a Wheeler NFA and we apply the powerset construction, in the worst case the equivalent DFA does not have $2^n - 1$ states, but only $2n - 1$ states. This paradigm was recently extended to *arbitrary* automata by introducing a new parameter p [5, 6, 7]. Informally speaking, an automaton has parameter p if we can partition the set of all states into p sets so that the convexity properties of Wheeler automata hold for all the edges leaving the same set and reaching the same set. In other words, p identified a relaxed notion of convexity. In the worst case, p is equal to the number n of states, and the case of Wheeler automata corresponds to $p = 1$. Unexpectedly, the results on Wheeler automata can be extended to arbitrary automata and, in particular, if we apply the powerset construction to an NFA with n states, the equivalent DFA has at most $2^p(n - p + 1) - 1$. In the worst case, $p = n$ and we retrieve the bound $2^n - 1$. For $p = 1$ (the Wheeler case), we retrieve the bound $2n - 1$. Most importantly, the quantity $2^p(n - p + 1) - 1$ is only exponential in p , so we can avoid the state explosion for small p (that is, if we have “quasi-convexity”).

Our goal is to extend the convexity paradigm to Petri nets. This is both of theoretical and practical interest. Currently, we do not know natural classes of automata for which p is – say – at most 5. At the same time, many Petri nets describing real systems have a small p – intuitively, many real systems are “quasi-convex”. For example, the Petri nets modeling the manufacturing system in Figure 2 can be divided into 4 linear components (one for each component and one describing the process after the merging). Every linear component can be seen as a total order, so it is intuitively reasonable to claim that p is at most 4 in our example.

We have three main objectives. First, we formalize the notion of convexity (or quasi-convexity) and we study the properties of convex Petri nets (boundness, reachability, and so on), focusing on

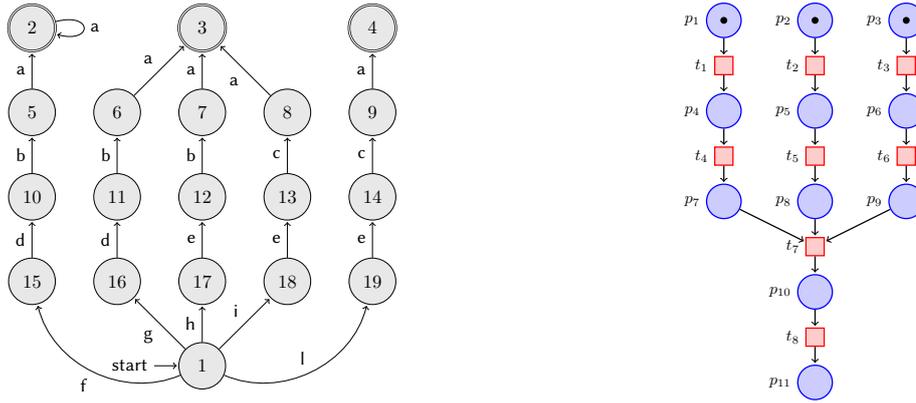


Figure 2: *Left:* A Wheeler NFA. *Right:* A Petri net modeling a manufacturing system where three distinct components are processed independently and then merged to obtain the final product (adapted from [8]).

the state explosion. Second, we show how convex Petri nets can model a broad spectrum of real systems by providing case studies. Lastly, we study the languages recognized by convex Petri nets. We know that finite automata recognize regular language (finite strings) and ω -regular languages (infinite strings). Wheeler automata recognize an interesting subclass of regular languages – *Wheeler languages* – that always admit a minimum Wheeler automata, an algebraic characterization, and a Myhill-Nerode theorem [3]. Petri nets can also be studied as acceptors of finite words [9] and infinite words [10], so the natural question is how to extend Wheeler languages to Petri nets by introducing the notion of convex (or Wheeler) Petri net language.

Acknowledgements. This work has been partially funded by the MUR-PRIN project DREAM (20228FT78M), and the MUR-PNRR project VITALITY (ECS00000041).

References

- [1] A. Valmari, The state explosion problem, in: *Advanced Course on Petri Nets*, Springer, 1996, pp. 429–528.
- [2] J. Esparza, M. Nielsen, Decidability issues for petri nets, *Petri nets newsletter* 94 (1994) 5–23.
- [3] J. Alanko, G. D’Agostino, A. Policriti, N. Prezsa, Regular languages meet prefix sorting, in: *Proceedings of the Thirty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA ’20*, Society for Industrial and Applied Mathematics, USA, 2020, p. 911–930.
- [4] Y. Peng, H. C. M. Leung, S. M. Yiu, F. Y. L. Chin, Idba – a practical iterative de bruijn graph de novo assembler, in: B. Berger (Ed.), *Research in Computational Molecular Biology*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 426–440.
- [5] N. Cotumaccio, G. D’Agostino, A. Policriti, N. Prezsa, Co-lexicographically ordering automata and regular languages - part i, *J. ACM* 70 (2023). URL: <https://doi.org/10.1145/3607471>. doi:10.1145/3607471.
- [6] N. Cotumaccio, N. Prezsa, On indexing and compressing finite automata, in: D. Marx (Ed.), *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, SIAM, 2021, pp. 2585–2599. URL: <https://doi.org/10.1137/1.9781611976465.153>. doi:10.1137/1.9781611976465.153.
- [7] N. Cotumaccio, Graphs can be succinctly indexed for pattern matching in $o(|e|^2 + |v|^{5/2})$ time, in: *2022 Data Compression Conference (DCC), 2022*, pp. 272–281. doi:10.1109/DCC52660.2022.00035.
- [8] R. Davidrajuh, *Petri nets for modeling of large discrete systems*, Springer, 2021.
- [9] J. L. Peterson, *Petri Net Theory and the Modeling of Systems*, Prentice Hall PTR, USA, 1981.
- [10] R. Valk, Infinite behaviour of petri nets, *Theoretical Computer Science* 25 (1983) 311–341. URL: <https://www.sciencedirect.com/science/article/pii/0304397583901159>. doi:[https://doi.org/10.1016/0304-3975\(83\)90115-9](https://doi.org/10.1016/0304-3975(83)90115-9).