Analysing Petri Nets with Higher Dimensional Automata

1 Context

Formal methods are mathematically rigorous techniques to ensure that a system design meets the given specification. There are numerous mathematical models, answering to different problems. These approaches reach from simple finite state machines to formal proof systems. In this internship, we are particularly interested in models for concurrent systems, that is to represent and analyze systems that can execute multiple actions at the same time, or out of order.
Among the existing models for concurrency, Petri Nets are of particular interest. They can model numerous semantics of concurrence by using a built-in notion of resources (tokens). Moreover, Petri Nets are an established standard in academia and industry as they have a nice graphical representation while retaining a high expressiveness. A more recent framework for modelling concurrency are Higher Dimensional Automata, which are an automaton based model representing both interleaving and non-interleaving concurrency.
As an example, consider Fig. 1 showing Petri Nets and HDA models for a system with two events, labelled a and b. The Petri Net and HDA on the left side model the (mutually exclusive) interleaving of a and b as either $a.b$ or $b.a$; those to the right model concurrent execution of a and b. In the HDA, this "non-interleaving independence" is indicated by a filled-in square.

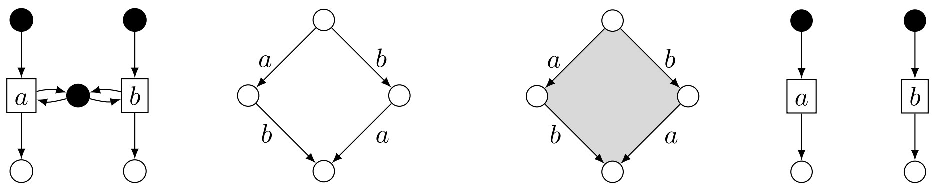

Figure 1: In Petri Nets, there are two types of "states": places (circles) and transitions (squares). A transition can be activated if all incoming places have a resource available (black, filled-in circle), destroying the incoming resource and producing a new resource on the outgoing places. On the left the Petri Net and HDA model for interleaving concurrency with the executions $a.b + b.a$ are shown. That is a and b can happen in any order, however, if we take take the actions to be noninstantaneous, they can never happen at the same time or overlap. On the right we have the models for "true", non-interleaving concurrency. Here a and b can, in addition to one after another, also happen at the same time or partially overlap, denoted as $a \parallel b$.

2 Objective of the internship

In recent years HDAs have become an increasingly active field of research, see [1, 2, 5, 6, 7] and references therein.

Indeed it was shown in [6] that HDA is a strictly more expressive model than most standard models of concurrency. That is, any instance of all these other models can in theory be translated to an HDA representing an equivalent behaviour. Despite this expressiveness and the significant theoretical advancements in recent years practical applications and implementations for HDAs are lacking.
The goal of this internship is to push forward practical applications for HDA, in particular techniques of Model Checking of Petri Nets by representing them as HDAs. The goal is to devise an efficient implementation for HDA allowing to take full advantage of the abstract representation of concurrency in HDAs. For instance, the only publicly available implementation for HDAs[1] and a preliminary proof of concept rely on an explicit representation of all cells of the HDA which necessarily leads to an exponential blow-up as the number of dimensions grows. To achieve this goal, new data structures and algorithms need to be devised.
Ideally, the implementation builds-on well known algorithms and tools for LTL model checking [3, 4] and will participate at the LTL track of the annual model checking contest[2].

3 Practical Informations

The internship is supervised by Hugo Bazille (Assistant professor Epita Rennes, LRE, Automata and Applications group) and Philipp Schlehuber-Caissier (Assistant professor Télécom SudParis, SAMOVAR, NeSS). It will by located at the Télécom SudParis campus (Rue Charles Fourier, Courcouronnes) with possible visits at Epita (Rue Voltaire, Le Kremlin-Bicêtre) and the campus at of Télécom SudParis in Palaiseau (Marguerite Perey, Palaiseau).
Mail contacts: hugo1.bazille@epita.fr, philipp.schlehuber-caissier@telecom-sudparis.eu

4 Qualifications

- Formal Methods and Languages
- C++

References

[1] Amazigh Amrane, Hugo Bazille, Emily Clement, and Uli Fahrenberg. Languages of higher-dimensional timed automata. In *International Conference on Applications and Theory of Petri Nets and Concurrency*, pages 197–219. Springer, 2024.

[2] Amazigh Amrane, Hugo Bazille, Uli Fahrenberg, and Marie Fortin. Logic and languages of higher-dimensional automata. In *International Conference on Developments in Language Theory*, pages 51–67. Springer, 2024.

[3] Christel Baier and Joost-Pieter Katoen. *Principles of model checking.* MIT press, 2008.

[4] Alexandre Duret-Lutz, Etienne Renault, Maximilien Colange, Florian Renkin, Alexandre Gbaguidi Aisse, Philipp Schlehuber-Caissier, Thomas Medioni, Antoine Martin, Jérôme Dubois, Clément Gillard, et al. From spot 2.0 to spot 2.10: what's new? In *International Conference on Computer Aided Verification*, pages 174–187. Springer, 2022.

[1] https://github.com/twkahl/PG2HDA
[2] https://mcc.lip6.fr/2024/

[5] Uli Fahrenberg, Christian Johansen, Georg Struth, and Krzysztof Ziemiański. Posets with interfaces as a model for concurrency. *Information and Computation*, 285:104914, 2022.

[6] Rob J van Glabbeek. On the expressiveness of higher dimensional automata. *Theoretical computer science*, 356(3):265–290, 2006.

[7] Safa Zouari, Krzysztof Ziemianski, and Uli Fahrenberg. Bisimulations and logics for higher-dimensional automata. *arXiv preprint arXiv:2402.01589*, 2024.