Category Theory for Compositional Verification

Kazuki Watanabe[1,2], Clovis Eberhart[2,3], Kazuyuki Asada[4], and Ichiro Hasuo[1,2]

[1] SOKENDAI (The Graduate University for Advanced Studies), Hayama, Japan
[2] National Institute of Informatics (NII), Tokyo, Japan
[3] Japanese-French Laboratory for Informatics, IRL 3527, Tokyo, Japan
[4] Tohoku University, Sendai, Japan

Introduction We present a recent line of work on *open games* and open structures [1,2,3], which describe compositional frameworks for parity games and Markov decision processes. This is done by adding *open ends* that form interfaces along which two structures can be composed. By finding adequate syntactic and semantic categories, we can then write algorithms that can leverage compositionality to compute faster solutions when there are repeated sub-structures.

Here, we show the categorical machinery at work in the definition of open parity games (OPGs) and related categories, as well as some particularities that arise in the case of open Markov decision processes (OMDPs). Our goal is to give all the necessary tools to understand the right-hand triangle in Figure 1 and why they are relevant. We also show that it is possible to derive efficient algorithms from category theory. Some details have been simplified for readability, for a full account see [1,2,3].

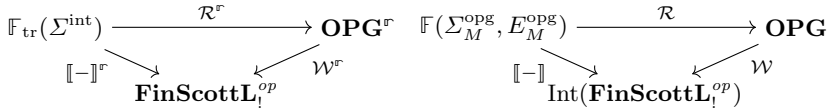

Fig. 1. The unidirectional and bidirectional frameworks (respectively left and right).

A Traced Symmetric Monoidal Category of OPGs We define a category $\mathbf{OPG}^{\mathbb{r}}$ of (rightward) OPGs whose objects are integers representing a number of open ends (which all point towards the right), and a morphism $n \to m$ is an OPG with n open ends on the left and m on the right. The identities, composition, unit, swaps, and tensor product are straightforward. Tracing is done by "rewiring" a number of open ends from the right interface into those of the left interface. This data can be shown to form a TSMC.

Graphical Language as a free TSMC However, manipulating OPGs can be cumbersome, especially because they need to be quotiented to form a category. Moreover, it is difficult to define algorithms that work by decomposing OPGs into sub-games. We therefore define a syntax for these games that is easier to handle. It is based on a signature Σ^{int} that contains a generator for each type

of node (i.e., tuple of player, parity, and left and right interfaces). It gives rise to the free TSMC $\mathbb{F}_{\mathrm{tr}}(\Sigma^{\mathrm{int}})$, and a *realisation functor* $\mathcal{R}^{\mathbb{r}} \colon \mathbb{F}_{\mathrm{tr}}(\Sigma^{\mathrm{int}}) \to \mathbf{OPG}^{\mathbb{r}}$ by mapping each generator to the corresponding one-node OPG. Fullness of $\mathcal{R}^{\mathbb{r}}$ can easily be shown, which shows that $\mathbb{F}_{\mathrm{tr}}(\Sigma^{\mathrm{int}})$ can represent all OPGs.

Semantic Category as a Kleisli category Since they are open, OPGs have positions that are neither winning nor losing because infinite plays may leave the OPG. To represent the semantics of OPGs, we use $\mathbf{FinScottL}_!^{op}$. Glossing over details, it is a TSMC whose represents choices by player and opponent of a path in the game from an entry to an exit, remembering the highest priority seen on the path. We can then define the *interpretation functor* $[\![-]\!]^{\mathbb{r}} : \mathbb{F}_{\mathrm{tr}}(\Sigma^{\mathrm{int}}) \to \mathbf{FinScottL}_!^{op}$ by freeness and the *winning-position functor* $\mathcal{W}^{\mathbb{r}} \colon \mathbf{OPG}^{\mathbb{r}} \to \mathbf{FinScottL}_!^{op}$ based on strategies in OPGs.

Bidirectionality and Int-construction We have built a triangle of TSMCs and TSMC functors as on the left of Figure 1. Using the Int-construction, we automatically get a triangle of compact closed categories and functors as on the right of Figure 1. This framework is simpler to use in practice, since open ends now have two directions (left and right), and loops can thus be created without having to explicitly use tracing. One nice point of our approach is that we get the full benefits of this more complex framework without having to go through complex definitions, simply by using the Int-construction.

Category Theory for Algorithms In [2], we have defined an algorithm to find the optimal expected reward in MDPs by induction on the syntax. The idea is to find all potentially-optimal schedulers in sub-MDPs, then compose them to find the optimal scheduler in the global MDP. This is done as above, by defining adequate syntactic and semantic categories, then using freeness of the syntactic category. This algorithm is faster than state of the art in case of repeated sub-MDPs, and its advantage grows stronger with the degree of repetition of sub-MDPs. In [3], we use the notion of Pareto curves to find a sound and efficient approximation of all schedulers for sub-MDPs.

References

1. Watanabe, K., Eberhart, C., Asada, K., Hasuo, I.: A compositional approach to parity games. In: Sokolova, A. (ed.) Proceedings 37th Conference on Mathematical Foundations of Programming Semantics, MFPS 2021, Hybrid: Salzburg, Austria and Online, 30th August - 2nd September, 2021. EPTCS, vol. 351, pp. 278–295 (2021). https://doi.org/10.4204/EPTCS.351.17
2. Watanabe, K., Eberhart, C., Asada, K., Hasuo, I.: Compositional probabilistic model checking with string diagrams of MDPs. In: International Conference on Computer Aided Verification. pp. 40–61. Springer (2023)
3. Watanabe, K., van der Vegt, M., Hasuo, I., Rot, J., Junges, S.: Pareto Curves for Compositionally Model Checking String Diagrams of MDPs. arXiv preprint arXiv:2401.08377 (2024)