Structure Meets Power Workshop

(Abstracts)

7 July 2024

https://www.cst.cam.ac.uk/conference/structure-meets-power-2024

Preface

This volume contains the papers presented at SmP 2024 Workshop: Structure Meets Power 2024 held July 7, 2024 in Tallinn and online.

The volume includes the abstracts of 7 accepted contributed talks and 2 invited talks. SmP 2024 is the fourth workshop in the series of Structure Meets Power workshops. Earlier instalments of the series have been held in 2021 (online, affiliated with LiCS 2021), 2022 (in Paris and online, affiliated with ICALP 2022) and 2023 (in Boston and online, affiliated with LiCS 2023).

The program of the SmP 2024 workshop focuses on bridging the divide in the field of logic in Computer Science, between two distinct strands: one focusing on semantics and compositionality ("Structure"), the other on expressiveness and complexity ("Power"). It is remarkable because these two fundamental aspects of our field are studied using almost disjoint technical languages and methods, by almost disjoint research communities. We believe that bridging this divide is a major issue in Computer Science, and may hold the key to fundamental advances in the field. The aim of the Structure meets Power workshop is to cultivate interaction between researchers who are interested in combining ideas from these two strands.

On July 2, 2024
in London, Cambridge, Nottingham, Oxford, and Prague.

Samson Abramsky
Anuj Dawar
Tomáš Jakl
Dan Marsden
Yoàv Montacute
Nihil Shah
Luca Reggio

Table of Contents

Where Büchi meets Parity

(INVITED TALK)

Karoliina Lehtinen

Abstract:

Automata over infinite objects are often defined either with a Büchi condition, which requires an accepting run to encounter something good infinitely often, or a Parity condition, which requires the most significance colour that occurs infinitely often on the run to be even.

Classic automata theory tells us that over infinite words, nondeterministic Büchi automata are as expressive as nondeterministic parity automata. Namely, both models recognise all ω-regular languages. Another by now classic result tells us that this is not the case over infinite trees, over which nondeterministic parity automata are much more expressive: unlike over words, no strictly weaker acceptance condition can recognise the same set of languages.

This raises the following question: over which classes of trees, beyond words, is the Büchi condition as expressive as the parity condition. In other words, when is Büchi enough?

In this talk, I will first take you on a tour of some of these classic results on the relative power of the Büchi and parity acceptance conditions, and how they relate to linear temporal logic and the modal μ-calculus. I will then revisit some of these methods and give a preview of fresh results characterising the classes of infinite trees over which Büchi automata are as expressive as parity automata. Finally, I will survey both long-standing and newer open problems on the power of there automata over different structures. Along the way, I will mention how these problems relate to the quest for a polynomial time algorithm for solving parity games.

Sweedler theory of monads

(INVITED TALK)

Tarmo Uustalu

Abstract:

Monad-comonad interaction laws are a mathematical concept for describing communication protocols between effectful computations and coeffectful environments in the paradigm where notions of effectful computation are modelled by monads and notions of coeffectful environment by comonads.

In this talk, I will demonstrate that monad-comonad interaction laws are an instance of measuring maps from López Franco and Vasilakopoulou's Sweedler theory for duoidal categories. The final interacting comonad for a monad and a residual monad arises as the Sweedler hom and the initial residual monad for a monad and an interacting comonad as the Sweedler copower.

I will explain a (co)algebraic characterization of monad-comonad interaction laws and how it leads to descriptions of the Sweedler hom and the Sweedler copower in terms of their coalgebras resp. algebras.

Joint work with Dylan McDermott and Exequiel Rivas.

A fibrational approach to regular languages of λ-terms

Vincent Moreau

IRIF, Université Paris Cité, Inria Paris, Paris, France
moreau@irif.fr

This is joint work with Paul-André Melliès.

Regular languages of λ-terms. There is a growing connection between automata theory and the theory of λ-calculus. Indeed, the Church encoding shows that finite words and ranked trees can be seen as simply typed λ-terms. For instance, words over the alphabet $\Sigma = \{a, b\}$ correspond to λ-terms of type

$$\mathsf{Church}_\Sigma \quad := \quad \underbrace{(\mathbb{o} \Rightarrow \mathbb{o})}_{a \text{ transition}} \Rightarrow \underbrace{(\mathbb{o} \Rightarrow \mathbb{o})}_{b \text{ transition}} \Rightarrow \underbrace{\mathbb{o}}_{\text{initial state}} \Rightarrow \underbrace{\mathbb{o}}_{\text{output state}}$$

Moreover, their semantic interpretations in the cartesian closed category **FinSet** coincides with their behavior in finite deterministic automata. This semantic observation led Salvati to define the notion of **recognizable language** in [5] as any set of λ-terms of a given type A of the form

$$\{M \text{ of type } A \mid [\![M]\!]_Q \in F\} \qquad \text{for some finite set } Q \text{ and subset } F \subseteq [\![A]\!]_Q.$$

The recognizable languages of type Church_Σ are then exactly the regular languages of words, seen through the Church encoding.

Logic in a bifibration. The idea that quantifiers are adjoints has been developed by Lawvere [3]. Inspired by [4], we follow this principle in its fibered form and reformulate Salvati's notion of regular language in terms of bifibrations preserving the CCC structure, highlighting the generality of the ingredients we use.

We write **SubSet** for the category whose objects are pairs (X, S) where X is a set and $S \subseteq X$, and whose morphisms are functions that restrict to the given subsets. The forgetful functor **SubSet** $\to$ **Set** is a bifibration, with the pullback being the inverse image and the pushforward being the direct image.

Moreover, **SubSet** is a CCC, with the internal hom being computed as

$$(X, S) \Rightarrow (Y, T) \quad := \quad (X \Rightarrow Y, \{f : X \to Y \text{ s.t. } \forall s \in S, f(s) \in T\})$$

and the bifibration **SubSet** $\to$ **Set** is a CCC functor. We write **Lam** for the category of types and simply typed λ-terms between them. We first ingredient is already known, see [2, Prop. 3.2].

Ingredient 1. *The pullback of a CCC bifibration along a cartesian functor is a CCC bifibration.*

We note $[\![A]\!]^\bullet_Q$ the subset of $[\![A]\!]_Q$ of points of the semantics which are λ-definable[1]. This assembles into a functor $[\![-]\!]^\bullet_Q : \mathbf{Lam} \to \mathbf{Set}$. By pullback, we hence obtain the bifibration

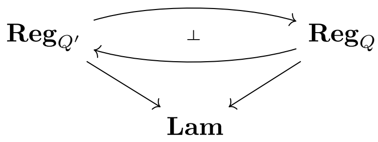
$$\begin{array}{ccc} \mathbf{Reg}_Q & \longrightarrow & \mathbf{SubSet} \\ \downarrow & \lrcorner & \downarrow \\ \mathbf{Lam} & \xrightarrow[{[\![-]\!]^\bullet_Q}]{} & \mathbf{Set} \end{array}$$

[1] $q \in [\![A]\!]_Q$ is λ-definable if there exists a λ-term M of type A such that $q = [\![M]\!]_Q$. This restriction on the subset $F \subseteq [\![A]\!]_Q$ removes useless recognizers and can be enforced without loss of generality, but the point of choosing λ-definable elements is really to be able to use Ingredient 2 later on.

More concretely, an object of $\mathbf{Reg}_Q$ is a pair (A, F) where A is a type and $F \subseteq [\![A]\!]^\bullet_Q$ is a subset of recognizers. As the functor $[\![-]\!]^\bullet_Q : \mathbf{Lam} \to \mathbf{Set}$ preserves cartesian products, we obtain that $\mathbf{Reg}_Q$ is a CCC and the bifibration $\mathbf{Reg}_Q \to \mathbf{Lam}$ is a CCC functor.

Ingredient 2. *A natural transformation between cartesian functors induces an adjunction between the respective pullbacks.*

We now explain how the different bifibrations $\mathbf{Reg}_Q$ are related. We make extensive use of the lemma of partial surjections [1, Prop. 3.2]. It implies that, for any finite sets Q and Q' such that Q' is of greater cardinality than Q, we have a natural transformation $\rho : [\![-]\!]^\bullet_{Q'} \Rightarrow [\![-]\!]^\bullet_Q$. We thus obtain the adjunction



obtained by pulling and pushing along ρ. The right adjoint represents the inclusion of Q-regular language into Q'-regular languages, and preserves pullbacks but not pushforwards. As such, it is merely a morphism of fibrations, not a morphism of bifibrations.

Ingredient 3. *The right adjoint induced by a natural transformation is a CCC functor if and only if the natural transformation satisfies Frobenius reciprocity.*

It follows from the general construction of Ingredient 2 that the functor $R : \mathbf{Reg}_Q \hookrightarrow \mathbf{Reg}_{Q'}$ preserves cartesian products. However, such a functor R does not preserve internal homs in general. Yet, the natural transformation ρ verifies the following Frobenius reciprocity condition over $\mathbf{Lam}$: for any $F \subseteq [\![A]\!]^\bullet_Q$ and $G \subseteq [\![A \Rightarrow B]\!]^\bullet_{Q'}$, the morphism

$$\exists_{\rho_A}(\exists_{\mathrm{ev}}(\rho_A^{-1}(F) \times \exists_{\mathrm{ap}}(G))) \quad \longrightarrow \quad \exists_{\mathrm{ev}}(F \times \exists_{\mathrm{ap}}(\exists_{\rho_{A \Rightarrow B}}(G)))$$

is an isomophism, where $\mathrm{ap} : [\![A \Rightarrow B]\!]^\bullet \to [\![A]\!]^\bullet \Rightarrow [\![B]\!]^\bullet$ is the canonical morphism obtained by currying the image of the evaluation. As a consequence, the functor $R : \mathbf{Reg}_Q \hookrightarrow \mathbf{Reg}_{Q'}$ is a CCC functor.

We define $\mathbf{Reg}$ as the colimit of all the $\mathbf{Reg}_Q$. It comes with a functor into $\mathbf{Lam}$, and as the inclusion functors $\mathbf{Reg}_Q \hookrightarrow \mathbf{Reg}_{Q'}$ preserve pullbacks and are CCC functors, we get that

$$\mathbf{Reg} \to \mathbf{Lam} \text{ is a fibration of CCCs.}$$

This generalizes some usual constructions of automata theory like the Brzozowski derivative.

Salvati's counterexample and MSO. The functor $\mathbf{Reg} \to \mathbf{Lam}$ has no reason to be an opfibration. This motivates us to give the following definition: a λ-term in $\mathbf{Lam}(A, B)$ will be said to preserve regular languages if, for any regular languages of type A, its image by the λ-term is a regular language of type B, or equivalently, i.e. if it has an opcartesian lifting to $\mathbf{Reg}$. The ability to push forward amounts to have an existential quantifier.

This is already witnessed at the level of trees: indeed, if we consider the types of trees

$$\begin{array}{rcl} A & := & (o \Rightarrow o) \Rightarrow (o \Rightarrow o) \Rightarrow o \Rightarrow o \\ B & := & (o \Rightarrow o \Rightarrow o) \Rightarrow (o \Rightarrow o) \Rightarrow o \Rightarrow o \end{array}$$

then the λ-term of type $A \Rightarrow B$ defined as $\lambda t.\lambda f.f\,(\lambda x.f\,x\,x)$ does not preserve regular languages. This corresponds to the fact that languages of trees are not closed by any homomorphic images, but only the linear ones[2]. Another, more involved example is given by Salvati in [5, §5.2].

However, some λ-terms preserve regular language. Indeed, the fact that any λ-term t of type A, seen as an element of $\mathbf{Lam}(1, A)$, preserves regular languages is exactly Statman's theorem, as the singleton language $\{t\}$ of type A is the pushforward along t of the singleton language of type 1.

In future work, we would like to find a sufficient condition for the preservation of regular languages, that would generalize Statman's theorem. This would constitute a first step towards a monadic second order logic for λ-terms.

References

[1] Sam van Gool, Paul-André Melliès, and Vincent Moreau. Profinite lambda-terms and parametricity. *Electronic Notes in Theoretical Informatics and Computer Science*, Volume 3 - Proceedings of MFPS XXXIX , November 2023.

[2] M. Hasegawa. Categorical glueing and logical predicates for models of linear logic. Preprint RIMS-1223, Kyoto University, 1999. `http://www.kurims.kyoto-u.ac.jp/~hassei/papers/full.pdf`.

[3] F. William Lawvere. Adjointness in foundations. *Dialectica*, 23:281–296, 1969.

[4] Paul-André Melliès and Noam Zeilberger. Functors are type refinement systems. In Sriram K. Rajamani and David Walker, editors, *Proceedings of the 42nd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2015, Mumbai, India, January 15-17, 2015*, pages 3–16. ACM, 2015.

[5] Sylvain Salvati. Recognizability in the simply typed lambda-calculus. In Hiroakira Ono, Makoto Kanazawa, and Ruy J. G. B. de Queiroz, editors, *Logic, Language, Information and Computation, 16th International Workshop, WoLLIC 2009, Tokyo, Japan, June 21-24, 2009. Proceedings*, volume 5514 of *Lecture Notes in Computer Science*, pages 48–60. Springer, 2009.

[2]The non-linearity is not between A and B, but inside of B

A Categorical Approach to Compositional Probabilistic Model Checking*

Kazuki Watanabe

National Institute of Informatics, Tokyo

1 Proposal

I would like to contribute a talk outlining our recent published works [21–23].

Systems with uncertainties, including *Markov decision processes (MDPs)* [17], are a main subject in verification. Markov decision processes have not only uncertainties induced by probabilistic transitions, but also non-determinisms induced by choices of probabilistic transitions. Choosing a *scheduler* on MDPs resolves a non-determinism, and optimal choices of schedulers are a central question in verification and planning. Specifically, a classical verification problem on MDPs is to find an optimal scheduler that resolves the non-determinism and maximizes the reachability probabilities or expected rewards until the target states are reached.

MDPs have also been actively studied in the context of computational effects and coalgebras. A seminal work by Varacca and Winskel [19] shows that there is no distributive law of the powerset monad over the distribution monad on the category of sets, the proof of which is credited to Gordon Plotkin. This result implies that a corresponding computational effect of MDPs is unlikely to have a monad structure, which raises a question about the compositionality of MDPs. There are several approaches to this problem: Varacca and Winskel [19] propose indexed valuations, Bonchi et al. [4] provide a trace semantics with the monad of convex subsets of distributions [16, 19], and Jacobs [13] provides an underlying theory of distributivities of the multiset monad over the distribution monad.

Compositionality is a fundamental property in category theory, and it has recently received attention in verification, including model checking. However, it has turned out to be challenging to obtain efficient compositional algorithms. Compositional probabilistic verification with respect to the product of MDPs has been proposed by Kwiatkowska et al. [15]. Since their framework is an assume-guarantee framework, it requires finding a suitable contract. Junges and Spaan [14] proposed a sequentially composed model checking algorithm with abstraction-refinement. A crucial assumption in [14] is the existence of optimal local sub-schedulers, which can be enforced to hold by restricting the form of sequential compositions.

Our recent series of works [21–23] addresses compositionality in probabilistic model checking with *string diagrams*. String diagrams are a celebrated graphical

* This talk is based on the joint works with Clovis Eberhart, Kazuyuki Asada, Ichiro Hasuo, Marck van der Vegt, Jurriaan Rot, and Sebastian Junges.

expression based on monoidal category theory. There are many successful applications of string diagrams, including quantum computing [2, 5], Petri nets [3, 18], ω-game [20], and games in economics [10]. The ultimate goal of our works [21–23] is to provide an efficient compositonal probabilistic model checking of MDPs with string diagrams, solving the notorious state space explosion problem.

In [21], we present the first compositional probabilistic model checking of MDPs with string diagrams. A crucial observation in [21] is that the *positional determinacy* of MDPs leads to a compositional reasoning. Positional determinacy is the property that says that positional schedulers do suffice. We then solve uncertainties on MDPs by the *change of enriching category* with the finite power set functor [6, 7]. We formalize our compositional probabilistic model checking by a compact closed functor.

In [22, 23], we develop practical approximation algorithms based on *Pareto curves* [8, 9]. We implemented our approximation algorithms in the model checker Storm [12], and demonstrate their performances compared to the state-of-the-art monolithic algorithm [11] and the exact compositional algorithm [21]. Unlike [21], the semantics in [22, 23] are not formulated in a categorical way. We conjecture that the semantics in [22, 23] can be interpreted as a Kleisli category of a certain "Pareto monad", which is mentioned in [1] as well.

References

1. Martín Abadi and Gordon D. Plotkin. Smart choices and the selection monad. *Log. Methods Comput. Sci.*, 19(2), 2023.
2. Samson Abramsky and Bob Coecke. A categorical semantics of quantum protocols. In *LICS*, pages 415–425. IEEE Computer Society, 2004.
3. Filippo Bonchi, Joshua Holland, Robin Piedeleu, Pawel Sobocinski, and Fabio Zanasi. Diagrammatic algebra: from linear to concurrent systems. *Proc. ACM Program. Lang.*, 3(POPL):25:1–25:28, 2019.
4. Filippo Bonchi, Ana Sokolova, and Valeria Vignudelli. The theory of traces for systems with nondeterminism, probability, and termination. *Log. Methods Comput. Sci.*, 18(2), 2022.
5. Bob Coecke and Aleks Kissinger. Picturing quantum processes - A first course on quantum theory and diagrammatic reasoning. In *Diagrams*, volume 10871 of *Lecture Notes in Computer Science*, pages 28–31. Springer, 2018.
6. Geoff SH Cruttwell. *Normed spaces and the change of base for enriched categories*. PhD thesis, Dalhousie University, 2008.
7. Samuel Eilenberg and G Max Kelly. Closed categories. In *Proceedings of the Conference on Categorical Algebra: La Jolla 1965*, pages 421–562. Springer, 1966.
8. Kousha Etessami, Marta Z. Kwiatkowska, Moshe Y. Vardi, and Mihalis Yannakakis. Multi-objective model checking of Markov decision processes. *Log. Methods Comput. Sci.*, 4(4), 2008.
9. Vojtech Forejt, Marta Z. Kwiatkowska, and David Parker. Pareto curves for probabilistic model checking. In *ATVA*, volume 7561 of *LNCS*, pages 317–332. Springer, 2012.
10. Neil Ghani, Jules Hedges, Viktor Winschel, and Philipp Zahn. Compositional game theory. In *LICS*, pages 472–481. ACM, 2018.

11. Arnd Hartmanns and Benjamin Lucien Kaminski. Optimistic value iteration. In *CAV (2)*, volume 12225 of *Lecture Notes in Computer Science*, pages 488–511. Springer, 2020.
12. Christian Hensel, Sebastian Junges, Joost-Pieter Katoen, Tim Quatmann, and Matthias Volk. The probabilistic model checker Storm. *Int. J. Softw. Tools Technol. Transf.*, 24(4):589–610, 2022.
13. Bart Jacobs. Partitions and ewens distributions in element-free probability theory. In *LICS*, pages 23:1–23:9. ACM, 2022.
14. Sebastian Junges and Matthijs T. J. Spaan. Abstraction-refinement for hierarchical probabilistic models. In *CAV (1)*, volume 13371 of *Lecture Notes in Computer Science*, pages 102–123. Springer, 2022.
15. Marta Z. Kwiatkowska, Gethin Norman, David Parker, and Hongyang Qu. Compositional probabilistic verification through multi-objective model checking. *Inf. Comput.*, 232:38–65, 2013.
16. Michael W. Mislove. Nondeterminism and probabilistic choice: Obeying the laws. In *CONCUR*, volume 1877 of *Lecture Notes in Computer Science*, pages 350–364. Springer, 2000.
17. Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* Wiley Series in Probability and Statistics. Wiley, 1994.
18. Julian Rathke, Pawel Sobocinski, and Owen Stephens. Compositional reachability in petri nets. In *RP*, volume 8762 of *Lecture Notes in Computer Science*, pages 230–243. Springer, 2014.
19. Daniele Varacca and Glynn Winskel. Distributing probability over nondeterminism. *Math. Struct. Comput. Sci.*, 16(1):87–113, 2006.
20. Kazuki Watanabe, Clovis Eberhart, Kazuyuki Asada, and Ichiro Hasuo. A compositional approach to parity games. In *MFPS*, volume 351 of *EPTCS*, pages 278–295, 2021.
21. Kazuki Watanabe, Clovis Eberhart, Kazuyuki Asada, and Ichiro Hasuo. Compositional probabilistic model checking with string diagrams of MDPs. In *CAV (3)*, volume 13966 of *LNCS*, pages 40–61. Springer, 2023.
22. Kazuki Watanabe, Marck van der Vegt, Ichiro Hasuo, Jurriaan Rot, and Sebastian Junges. Pareto curves for compositionally model checking string diagrams of MDPs. In *TACAS (2)*, volume 14571 of *LNCS*, pages 279–298. Springer, 2024.
23. Kazuki Watanabe, Marck van der Vegt, Sebastian Junges, and Ichiro Hasuo. Compositional value iteration with Pareto caching, 2024. to appear in CAV2024.

The free bifibration over a functor

Bryce Clarke & Gabriel Scherer & Noam Zeilberger

(Abstract submitted to Structure Meets Power 2024, May 16, 2024)

A functor $p : \mathcal{D} \to \mathcal{C}$ between two categories is a *bifibration* when, roughly speaking, objects of $\mathcal{D}$ may be pushed and pulled along arrows of $\mathcal{C}$. Formally, for any arrow $f : A \to B$ in $\mathcal{C}$ and any object S in $\mathcal{D}$ such that $p(S) = A$, there should be an object $f_* S$ and an arrow $f_S : S \to f_* S$ of $\mathcal{D}$ such that $p(f_S) = f$,

$$\begin{array}{ccc} \mathcal{D} & & S \overset{f_S}{\dashrightarrow} f_* S \\ {\scriptstyle p}\downarrow & & \\ \mathcal{C} & & A \xrightarrow{f} B \end{array}$$

which are universal in the sense that for any arrow $g : B \to C$ in $\mathcal{C}$ and arrow $\alpha : S \to T$ in $\mathcal{D}$ such that $p(\alpha) = fg$, there is a unique arrow $\beta : f_* S \to T$ such that $\alpha = f_S\, \beta$.

$$\begin{array}{c} S \xrightarrow{\quad\alpha\quad} T \\ \\ A \xrightarrow{f} B \xrightarrow{g} C \end{array} \quad = \quad \begin{array}{c} S \xrightarrow{f_S} f_* S \overset{\beta}{\dashrightarrow} T \\ \\ A \xrightarrow{f} B \xrightarrow{g} C \end{array} \tag{1}$$

Dually, for any arrow $g : B \to C$ in $\mathcal{C}$ and object T in $\mathcal{D}$ such that $p(T) = C$, there should be an object $g^* T$ and an arrow $\bar{g}_T : g^* T \to T$ of $\mathcal{D}$ such that $p(\bar{g}_T) = g$,

$$\begin{array}{c} g^* T \overset{\bar{g}_T}{\dashrightarrow} T \\ \\ B \xrightarrow{g} C \end{array}$$

again universal in the sense that for any arrow $f : A \to B$ in $\mathcal{C}$ and arrow $\alpha : S \to T$ in $\mathcal{D}$ such that $p(\alpha) = fg$, there is a unique arrow $\beta : S \to g^* T$ such that $\alpha = \beta\, \bar{g}_T$.

$$\begin{array}{c} S \xrightarrow{\quad\alpha\quad} T \\ \\ A \xrightarrow{f} B \xrightarrow{g} C \end{array} \quad = \quad \begin{array}{c} S \overset{\beta}{\dashrightarrow} g^* T \xrightarrow{\bar{g}_T} T \\ \\ A \xrightarrow{f} B \xrightarrow{g} C \end{array} \tag{2}$$

An immediate consequence of the definition is that if $p : \mathcal{D} \to \mathcal{C}$ is a bifibration then the operations of pushing or pulling along an arrow $f : A \to B$ of $\mathcal{C}$ extend to a pair of adjoint functors

$$\mathcal{D}_A \underset{f^*}{\overset{f_*}{\rightleftarrows}} \mathcal{D}_B \qquad (f_* \dashv f^*)$$

where $\mathcal{D}_A$ and $\mathcal{D}_B$ are the *fiber categories* defined as the subcategories of $\mathcal{D}$ spanned by the arrows living over the identities id_A and id_B in $\mathcal{C}$, and indeed any (cloven) bifibration over $\mathcal{C}$ may be equivalently described by the data of a pseudofunctor $\mathcal{C} \to \mathcal{A}\mathrm{dj}$ into the category of small categories and adjunctions.

The categorical notion of bifibration was originally introduced by Grothendieck, together with the weaker notion of fibration where one only has the ability to pull objects of the category above along arrows of the category below. One reason for the special interest of bifibrations from the perspective of logic and computer science is that the operations of pushing forward or pulling back along an arrow may be seen as generalizations of existential and universal quantification (cf. [MZ16]), and hence by alternating these operations one can in some sense define objects of arbitrary quantifier complexity. The pushforward and pullback operations may also be seen as generalizations of strongest postconditions and weakest preconditions in specification logics.

Although most functors are not bifibrations, any functor $p : \mathcal{D} \to \mathcal{C}$ generates a *free bifibration*, in the sense that there is a (cloven) bifibration $\Lambda_p : \mathcal{B}\mathrm{if}(p) \to \mathcal{C}$ and a functor $\eta_p : \mathcal{D} \to \mathcal{B}\mathrm{if}(p)$ such that $p = \Lambda_p \circ \eta_p$. Moreover, the free bifibration is universal in the sense that if $q : \mathcal{E} \to \mathcal{C}$ is any bifibration equipped with a functor $\theta : \mathcal{D} \to \mathcal{E}$ such that $p = q \circ \theta$, then there is an essentially unique morphism of bifibrations $\tilde{\theta} : \mathcal{B}\mathrm{if}(p) \to \mathcal{E}$ such that $\theta = \tilde{\theta} \circ \eta_p$. Whereas the free fibration over a functor has a well-known and very simple concrete description, the free bifibration has been relatively little studied, and describing it explicitly is far more subtle. The problem of building the free bifibration over a functor $p : \mathcal{D} \to \mathcal{C}$ is closely related to the problem, studied by Dawson, Paré, and Pronk [DPP03a, DPP03b], of extending $\mathcal{C}$ to a 2-category $\Pi_2\mathcal{C}$ by freely adjoining right adjoints (cf. [SS86]). However, as far as we are aware there is only one direct construction of the free bifibration over a functor in the literature, by Lamarche [Lam10, Lam14], Moreover, it should be said that both constructions (and the proofs of their correctness) involve a significant degree of combinatorial intricacy.

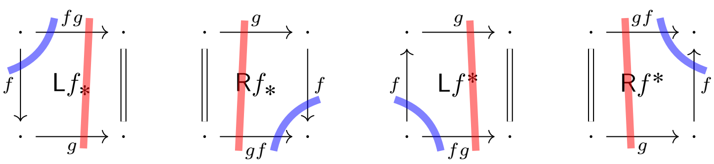
In our work, we have developed a number of alternative descriptions of the free bifibration over an arbitrary functor $p : \mathcal{D} \to \mathcal{C}$. One description is proof-theoretic, viewing the objects of $\mathcal{B}\mathrm{if}(p)$ as formulas in a primitive logic containing unary connectives f_* and f^* for every morphism f of $\mathcal{C}$, with the objects of $\mathcal{D}$ serving as atomic formulas. The morphisms of $\mathcal{B}\mathrm{if}(p)$ are then defined as equivalence classes of proofs in a simple cut-free sequent calculus containing only four logical rules

$$\frac{S \Longrightarrow_{fg} T}{f_* S \Longrightarrow_g T}\ \mathsf{L}f_* \qquad \frac{S \Longrightarrow_g T}{S \Longrightarrow_{gf} f_* T}\ \mathsf{R}f_* \qquad \frac{S \Longrightarrow_g T}{f^* S \Longrightarrow_{fg} T}\ \mathsf{L}f^* \qquad \frac{S \Longrightarrow_{gf} T}{S \Longrightarrow_g f^* T}\ \mathsf{R}f^*$$

where proofs are considered modulo four permutation relations, including the relations

$$\dfrac{\dfrac{S \Longrightarrow_{fh} T}{S \Longrightarrow_{fhg} g_* T}\ \mathsf{R}g_*}{f_* S \Longrightarrow_{hg} g_* T}\ \mathsf{L}f_* \sim \dfrac{\dfrac{S \Longrightarrow_{fh} T}{f_* S \Longrightarrow_h T}\ \mathsf{L}f_*}{f_* S \Longrightarrow_{hg} g_* T}\ \mathsf{R}g_* \qquad \dfrac{\dfrac{S \Longrightarrow_{h} T}{S \Longrightarrow_{hg} g_* T}\ \mathsf{R}g_*}{f^* S \Longrightarrow_{fhg} g_* T}\ \mathsf{L}f^* \sim \dfrac{\dfrac{S \Longrightarrow_{h} T}{S^* \Longrightarrow_{fh} T}\ \mathsf{L}f^*}{f^* S \Longrightarrow_{fhg} g_* T}\ \mathsf{R}g_*$$

as well as their symmetric versions with pushforward and pullback swapped. The cut rule is admissible, thereby defining composition of morphisms in $\mathcal{B}\mathrm{if}(p)$. This sequent calculus is closely related to an alternative description of the free bifibration using double category theory. The *double category of zigzags* $\mathbb{Z}\mathcal{C}$ has objects and horizontal arrows given by the objects and arrows of $\mathcal{C}$, vertical arrows given by zigzags (= signed sequences of arrows) in $\mathcal{C}$, and double cells of *zigzag morphisms* generated by vertical pastings of the four generating cells below (ignore the colored arcs for now)



modulo four permutation relations. Composition of zigzag morphisms can be defined inductively by analysis of the intermediate zigzag, thereby defining horizontal composition for the double category. The connection with bifibrations is that $\mathbb{Z}\mathcal{C}$ is the free bifibration over the identity functor $\mathrm{id}_{\mathcal{C}} : \mathcal{C} \to \mathcal{C}$, while conversely, any free bifibration may be reconstructed by pulling back the source functor $\mathrm{src} : \mathbb{Z}\mathcal{C} \to \mathcal{C}$ of the double category along an arbitrary functor $p : \mathcal{D} \to \mathcal{C}$. Finally, zigzag morphisms in $\mathbb{Z}\mathcal{C}$ also have a natural graphical interpretation (at least in the case where $\mathcal{C}$ is a free category) as certain planar arc diagrams considered up to isotopy (see colored arcs above).

A challenge in understanding free bifibrations is getting a handle on the equivalence classes of zigzag morphisms induced by the permutation relations. Indeed, by a reduction of [DPP03b], this equivalence relation is in general undecidable! One way we have attacked the problem is via the proof-theoretic technique of *focusing,* developing a (for now conjectured) canonical form whereby permutation equivalence classes of derivations in the above sequent calculus are represented by focused derivations modulo a much more rigid notion of equivalence. When $\mathcal{C}$ is *factorization preordered* this notion of equivalence reduces to equality, yielding a new proof of the decidability of zigzag morphism equality in that case, as first shown by [DPP03b].

Another way we have attacked the problem is by considering examples, and here is where it appears that free bifibrations give rise to a number of categories of great combinatorial interest. A basic example is the free bifibration over the functor $p = (* \mapsto 0) : 1 \to 2$ sending the unique object of 1 to the initial object of the walking arrow category $2 = \{ 0 \to 1 \}$. In this case, objects in the fiber over 0 are isomorphic to even-length sequences of alternating pushes and pulls $f^* f_* \cdots f^* f_* 0$ along the unique arrow $f : 0 \to 1$, while objects in the fiber over 1 correspond to odd-length sequences $f_* f^* f_* \cdots f^* f_* 0$. When we consider morphisms, it turns out that the fiber category $\mathcal{B}\mathrm{if}(p)_0$ is equivalent to the (augmented) simplex category Δ of finite ordinals and order-preserving maps, under an interpretation reading the length $2n$ sequence $f^* f_* \cdots f^* f_* 0$ as the ordinal $\mathsf{n} = \{ 0 < 1 < \cdots < n-1 \}$. Similarly, the fiber $\mathcal{B}\mathrm{if}(p)_1$ is equivalent to the category $\Delta_\perp$ of finite non-empty ordinals and order-and-least-element-preserving maps. In particular, from the sequent calculus for free bifibrations we can easily derive the well-known formula $\binom{n+m-1}{m}$ for the number of maps $\mathsf{m} \to \mathsf{n}$ in Δ. It is also worth mentioning that in this case the total category of the free bifibration is equivalent $\mathcal{B}\mathrm{if}(p) \cong \Upsilon$ to the *category of schedules* Υ introduced by Harmer, Hyland, and Melliès [HHM07] in their study of the categorical combinatorics of game semantics.

An even richer structure emerges considering the free bifibration over the functor $p = (* \mapsto 0) : 1 \to \mathbb{N}$ sending the unique object of 1 to the initial object of the natural numbers considered as a posetal category under the natural order. In this case, objects in the fiber of 0 are isomorphic to sequences of rising and falling steps in $\mathbb{N}$ that start at 0 and end at 0. In other words, they correspond to Dyck paths! By the standard bijection between Dyck paths and rooted planar trees, the fiber $\mathcal{B}\mathrm{if}(p)_0$ may therefore be interpreted as a category of trees, giving rise to an interesting notion of *morphism of planar trees*. Indeed, it turns out that $\mathcal{B}\mathrm{if}(p)_0$ is equivalent to a category of finite rooted planar trees that was defined in an entirely different manner by Joyal [Joy97] and Batanin [Bat98], namely as the full subcategory of the functor category $[\mathbb{N}^{op}, \Delta]$ consisting of those functors $T : \mathbb{N}^{op} \to \Delta$ such that $T(0) = \mathsf{1}$ and such that $T(h) = \mathsf{0}$ for some h. Under the Joyal-Batanin representation of planar trees, the ordinal $T(n)$ counts the number of nodes of height n from the root, while the monotone functions $T(n+1) \to T(n)$ map the nodes of height $n+1$ to their parent nodes of height n (these functions are necessarily order-preserving by planarity). It turns out that natural transformations between such functors are in one-to-one correspondence with equivalence classes of zigzag morphisms between the corresponding Dyck paths. In particular, we can enumerate natural transformations between trees by enumerating focused derivations in the sequent calculus for the free bifibration. Finally, it appears that we get some interesting combinatorics by fixing a tree T and considering the sequences

$$in[T]_n = \#\{ \alpha : T' \to T \mid |T'| = n \} \qquad out[T]_n = \#\{ \alpha : T \to T' \mid |T'| = n \}$$

counting all of the morphisms into T or out of T and out of/into a tree of a given size.

References

[Bat98] Michael Batanin. Monoidal globular categories as a natural environment for the theory of weak n-categories. *Advances in Mathematics*, 136:39–103, 1998. doi:10.1006/aima.1998.1724.

[DPP03a] Robert Dawson, Robert Paré, and Dorette Pronk. Adjoining adjoints. *Advances in Mathematics*, 178(1):99–140, 2003. doi:10.1016/S0001-8708(02)00068-3.

[DPP03b] Robert Dawson, Robert Paré, and Dorette Pronk. Undecidability of the free adjoint construction. *Applied Categorical Structures*, 11:403–419, 2003. doi:10.1023/A:1025712521140.

[HHM07] Russell Harmer, Martin Hyland, and Paul-André Melliès. Categorical combinatorics for innocent strategies. In *22nd IEEE Symposium on Logic in Computer Science (LICS 2007), 10-12 July 2007, Wroclaw, Poland, Proceedings*, pages 379–388. IEEE Computer Society, 2007. doi:10.1109/LICS.2007.14.

[Joy97] André Joyal. Disks, duality, and θ-categories. preprint, 1997. URL: https://ncatlab.org/nlab/files/JoyalThetaCategories.pdf.

[Lam10] François Lamarche. Path functors in Cat. preprint, 2010. URL: https://hal.inria.fr/hal-00831430.

[Lam14] François Lamarche. Modeling Martin-Löf type theory in categories. *Journal of Applied Logic*, 12:28–44, 2014. doi:10.1016/j.jal.2013.08.003.

[MZ16] Paul-André Melliès and Noam Zeilberger. A bifibrational reconstruction of Lawvere's presheaf hyperdoctrine. In *Proceedings of the 31st Annual IEEE Conference on Logic in Computer Science*, pages 555–564, July 2016.

[SS86] Stephen Schanuel and Ross Street. The free adjunction. *Cahiers de topologie et géométrie différentielle catégoriques*, 27(1):81–83, 1986. URL: http://www.numdam.org/item/CTGDC_1986__27_1_81_0/.

Context-Free Languages of String Diagrams

Extended Abstract*

Matt Earnshaw (Tallinn University of Technology)
j.w.w. Mario Román (University of Oxford)

May 17, 2024

This work introduces context-free languages of morphisms in monoidal categories, extending recent work on regular languages of string diagrams and the categorification of context-free languages. Context-free languages of string diagrams include classical context-free languages of words, trees, and hypergraphs, when instantiated over appropriate monoidal categories.

Context-free languages over free monoids, and other categories. In recent work, Melliès and Zeilberger [7, 8] pursue a structural approach to context-free grammars in terms of the "multiple-input, single-output" *multigraphs* underlying *multicategories*. This leads to a notion of context-free language of *morphisms* in a category, which recovers various generalizations of context-free languages. The authors prove a generalized Chomsky-Schützenberger representation theorem [1] which hinges on an adjunction between *contouring multicategories* and *splicing morphisms in a category* [8]. We start by recalling the latter, which sets up the definition of context-free grammar over a category.

Definition 1 (Melliès and Zeilberger [8]). *The* multicategory of spliced arrows*, $\mathscr{W}\mathbb{C}$, over a category $\mathbb{C}$, has, as objects, pairs of objects of $\mathbb{C}$, denoted $\frac{A}{B}$. Its multimorphisms are morphisms of the original category, but with n "gaps" or "holes". More precisely, the multimorphisms of $\mathscr{W}\mathbb{C}$ are given by:*

$$\mathscr{W}\mathbb{C}\left(\tfrac{A_1}{B_1}, \ldots, \tfrac{A_n}{B_n}; \tfrac{X}{Y}\right) := \mathbb{C}(X; A_1) \times \prod_{i=1}^{n-1} \mathbb{C}(B_i; A_{i+1}) \times \mathbb{C}(B_n; Y).$$

By convention, nullary multimorphisms are morphisms of $\mathbb{C}$, that is $\mathscr{W}\mathbb{C}(; \tfrac{X}{Y}) := \mathbb{C}(X; Y)$. The identity is given by a pair of identities of the original category, multicategorical composition is derived from the composition in the original category, by "splicing" into holes.

Definition 2 (Melliès and Zeilberger [8]). *A context-free grammar of morphisms in a category $\mathbb{C}$ is a morphism of multigraphs $\phi : G \to |\mathscr{W}\mathbb{C}|$, where G is finite, and a start symbol S in G. The language of the grammar is the image of the set of multimorphisms $\mathcal{F}_{\triangledown}G(; S)$ in the free multicategory over G under the multifunctor $\mathcal{F}_{\triangledown}\phi$, which by Definition 1 is a set of morphisms in $\mathbb{C}$.*

Context-free grammars over a monoidal category. For categories equipped with a monoidal structure, it is natural to consider more general kinds of holes than permitted by the spliced arrows construction, as illustrated in Figure 1. We call these more general morphisms with holes *diagram contexts*. Diagram contexts assemble into a symmetric multicategory.

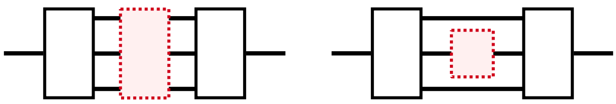


Figure 1: (Left) A spliced morphism is a tuple of morphisms. (Right) In a monoidal category we may have more general holes, which do not split a morphism into disjoint pieces.

*This extended abstract is based on the preprint *Context-Free Languages of String Diagrams* [2].

Definition 3. *The symmetric multicategory of diagram contexts* $\boxed{\mathcal{P}}$ *over a monoidal category* $\mathcal{P}$*, has multimorphisms given by derivable sequents in the following type theory, where* X, Y, Z, X_i, Y_i *are objects of* $\mathcal{P}$ *and* $f : X_1 \otimes ... \otimes X_n \to Y_1 \otimes .... \otimes Y_m$ *is a morphism of* $\mathcal{P}$*.*

$$\frac{}{\vdash \mathrm{id} : {}^{X}_{X}} \qquad \frac{}{\vdash f : {}^{X_1,\dots,X_n}_{Y_1,\dots,Y_m}} \qquad \frac{}{\boxed{x} : {}^{X}_{Y} \vdash \boxed{x} : {}^{X}_{Y}}$$

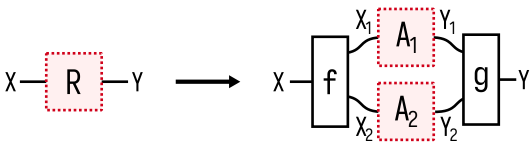
$$\frac{\Gamma \vdash t_1 : {}^{X}_{Y} \quad \Delta \vdash t_2 : {}^{Y}_{Z}}{\mathrm{Shuf}(\Gamma;\Delta) \vdash t_1 \mathbin{⨾} t_2 : {}^{X}_{Z}} \qquad \frac{\Gamma \vdash t_1 : {}^{X_1}_{Y_1} \quad \Delta \vdash t_2 : {}^{X_2}_{Y_2}}{\mathrm{Shuf}(\Gamma;\Delta) \vdash t_1 \otimes t_2 : {}^{X_1 +\!+ X_2}_{Y_1 +\!+ Y_2}}$$

We consider terms up to α*-equivalence and we impose the following equations over the terms whenever they are constructed over the same context:* $(t_1 ⨾ t_2) ⨾ t_3 = t_1 ⨾ (t_2 ⨾ t_3)$; $t ⨾ \mathrm{id} = t$; $t_1 \otimes (t_2 \otimes t_3) = (t_1 \otimes t_2) \otimes t_3$; $(t_1 ⨾ t_2) \otimes (t_3 ⨾ t_4) = (t_1 \otimes t_3) ⨾ (t_2 \otimes t_4)$.

We can now replace *spliced arrows* over a category in Definition 2 by *diagram contexts* in a monoidal category, to obtain a notion of context-free grammar of morphisms in a monoidal category. We also consider morphisms of *symmetric* multigraphs, as holes in a diagram context have no canonical ordering.

Definition 4. *A context-free monoidal grammar over a strict monoidal category* $(\mathbb{C}, \otimes, I)$ *is a morphism of symmetric multigraphs* $\Psi : \mathcal{G} \to |\boxed{\mathbb{C}}|$*, into the underlying multigraph of diagram contexts in* $\mathbb{C}$*, where* $\mathcal{G}$ *is finite, and a start symbol* $S_{X,Y} \in \Psi^{-1}({}^{X}_{Y})$*. The language of the grammar is a set of morphisms in* $\mathbb{C}(X;Y)$*, defined analogously to Definition 2.*

Context-free monoidal grammars admit a convenient diagrammatic presentation using string diagrams for monoidal categories. For example, given a multimorphism $r \in \mathcal{G}(A_1, A_2; R)$ with image $\boxed{a} : {}^{X_1}_{Y_1}, \boxed{b} : {}^{X_2}_{Y_2} \vdash f ⨾ \boxed{a} \otimes \boxed{b} ⨾ g : {}^{X}_{Y} \in |\boxed{\mathbb{C}}|({}^{X_1}_{Y_1}, {}^{X_2}_{Y_2}; {}^{X}_{Y})$, we draw the following:



Example 1. Context-free tree grammars. *Context-free tree grammars* [6, 10] are equivalent to context-free monoidal grammars over the free cartesian monoidal category on a signature of terminals (e.g. Figure 2).

Example 2. Hypergraph grammars. Hyperedge-replacement (HR) grammars are a kind of *context-free graph grammar* [5]. Hypergraphs are the morphisms of *hypergraph* monoidal categories [9]. Generators in a monoidal signature are directed hyperedges, and the extra structure in a hypergraph category amounts to a combinatorial encoding of patterns of wiring. Let Γ be a monoidal signature of terminal hyperedges, G a finite multigraph of non-terminals, and $S \in G$ a start symbol. Then context-free monoidal grammars over the free hypergraph category $\mathsf{Hyp}[\Gamma]$ are exactly hyperedge replacement grammars over Γ.

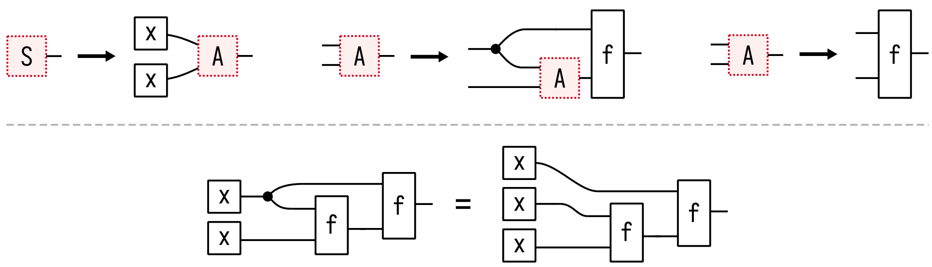


Figure 2: Example of a context-free tree grammar as a monoidal grammar. The string diagrams at the bottom are *equal* in the free *cartesian* category over the terminals.

Optical contour of a multicategory. An important contribution of Melliès and Zeilberger is their exhibition of a left adjoint to the formation of spliced arrows in a category (Definition 1); the *contour* of a multicategory [8, Section 3.2]. This adjunction is a key tool in their generalized Chomsky-Schützenberger theorem. A similar adjunction has been introduced by the Earnshaw, Hefford and Román between produoidal categories and monoidal categories [3]. Here, we introduce a version of the latter, between the *optical contour* of a multicategory and *raw optics* in a monoidal category. A raw optic is a tuple of morphisms obtained by cutting a diagram context into a tuple.

Definition 5. *The multicategory of raw optics over a strict monoidal category* $\mathbb{C}$*, denoted* $\mathsf{ROpt}[\mathbb{C}]$*, is defined to have, as objects, pairs* $\frac{A}{B}$ *of objects of* $\mathbb{C}$*. Its multimorphisms* $\mathsf{ROpt}[\mathbb{C}](\frac{A_1}{B_1},\dots,\frac{A_n}{B_n};\frac{S}{T})$ *are*

$$\sum_{M_i,N_i\in\mathbb{C}} \mathbb{C}(S; M_1\otimes A_1\otimes N_1) \times \prod_{i=1}^{n-1} \mathbb{C}(M_i\otimes B_i\otimes N_i; M_{i+1}\otimes A_{i+1}\otimes N_{i+1}) \times \mathbb{C}(M_n\otimes B_n\otimes N_n; T).$$

As a special case, $\mathsf{ROpt}[\mathbb{C}](;\frac{S}{T}) := \mathbb{C}(S;T)$*. Identities are given by pairs* $(\mathsf{id}_A, \mathsf{id}_B)$*, and composition by splicing pieces together.*

Note that raw optics are not spliced arrows in the sense of Melliès and Zeilberger: the objects M_i, N_i must "match" in adjacent pieces. Raw optics has a left adjoint, the *optical contour*. This is similar to the contour operation introduced by Melliès and Zeilberger but with additional objects M_i, N_i, introduced to keep track of strings that surround holes. This gives rise to a strict monoidal category.

Theorem 1. *Optical contour is left adjoint to raw optics:* $(\mathcal{C} \dashv \mathsf{ROpt}) : \mathit{MonCat} \to \mathit{MultiCat}$.

A monoidal representation theorem. The Chomsky-Schützenberger representation theorem says that every context-free language can be obtained as the image under a homomorphism of the intersection of a Dyck language and a regular language [1]. Melliès and Zeilberger [7] use their splicing-contour adjunction to give a novel proof of this theorem for context-free languages in categories. The role of the Dyck language is taken over by *contours* of derivations.

Following recent work by Sobociński and the first author on *regular* languages of string diagrams [4], we investigate the representation theorem for context-free languages of string diagrams. In this case, the optical contour of a grammar is already a *regular* language of string diagrams, and this is sufficient to reconstruct the original language, with the help of a monoidal functor.

Theorem 2. *Every context-free language of string diagrams is the image under a monoidal functor of a regular language of string diagrams.*

References

[1] Noam Chomsky and Marcel-Paul Schützenberger. The algebraic theory of context-free languages. In P. Braffort and D. Hirschberg, editors, *Computer Programming and Formal Systems*, volume 35 of *Studies in Logic and the Foundations of Mathematics*, pages 118–161. Elsevier, 1963.

[2] Matt Earnshaw and Mario Román. Context-free languages of string diagrams, 2024. `arXiv:2404.10653`.

[3] Matthew Earnshaw, James Hefford, and Mario Román. The Produoidal Algebra of Process Decomposition. In Aniello Murano and Alexandra Silva, editors, *CSL 2024*, volume 288 of *LIPIcs*, pages 25:1–25:19, 2024. `doi:10.4230/LIPIcs.CSL.2024.25`.

[4] Matthew Earnshaw and Pawel Sobociński. Regular planar monoidal languages. *Journal of Logical and Algebraic Methods in Programming*, 2024.

[5] Joost Engelfriet. *Context-Free Graph Grammars*, pages 125–213. Springer Berlin Heidelberg, Berlin, Heidelberg, 1997. `doi:10.1007/978-3-642-59126-6_3`.

[6] Ferenc Gécseg and Magnus Steinby. *Tree Languages*, page 1–68. Springer-Verlag, 1997.

[7] Paul-André Melliès and Noam Zeilberger. Parsing as a Lifting Problem and the Chomsky-Schützenberger Representation Theorem. In *MFPS 2022-38th conference on Mathematical Foundations for Programming Semantics*, 2022.

[8] Paul-André Melliès and Noam Zeilberger. The categorical contours of the Chomsky-Schützenberger representation theorem. Preprint, December 2023.

[9] Robert Rosebrugh, Nicoletta Sabadini, and Robert F.C. Walters. Generic commutative separable algebras and cospans of graphs. *Theory and applications of categories*, 15:164–177, 2005.

[10] William C. Rounds. Context-free grammars on trees. In *Proceedings of the First Annual ACM Symposium on Theory of Computing*, STOC '69, page 143–148, New York, NY, USA, 1969. Association for Computing Machinery. `doi:10.1145/800169.805428`.

Unifying algebraic lower bounds, semantically

(Joint work with L. Pellissier and U. Léchine [26])

Thomas Seiller
CNRS
thomas.seiller@cnrs.fr

May 17, 2024

1 Introduction

Lower bounds. Complexity theory has traditionally been concerned with proving *separation results*. Among the numerous open separation problems lies the much advertised Ptime vs. NPtime problem of showing that some problems considered hard to solve but efficient to verify do not have a polynomial time algorithm solving them.

Proving that two classes $B \subset A$ are not equal can be reduced to finding lower bounds for problems in A: by proving that certain problems cannot be solved with less than certain resources on a specific model of computation, one can show that two classes are not equal. Conversely, proving a separation result $B \subsetneq A$ provides a lower bound for the problems that are A*-complete* [8] – i.e. problems that are in some way *universal* for the class A.

Alas, the proven lower bound results are very few, and most separation problems remain as generally accepted conjectures. For instance, a proof that the class of non-deterministic exponential problems is not included in what is thought of as a very small class of circuits was not achieved until very recently [29].

The failure of most techniques of proof has been studied in itself, which lead to proofs of negative results commonly called *barriers*. Altogether, these results show that all proof methods we know are ineffective with respect to proving interesting lower bounds. Indeed, there are three barriers: relativisation [6], natural proofs [19] and algebrization [1], and every known proof method hits at least one of them. This shows the need for new methods[1]. However, to this day, only one research program aimed at proving new separation results is commonly believed to have the ability to bypass all barriers: Mulmuley and Sohoni's Geometric Complexity Theory (GCT) program [17].

Geometric Complexity Theory is widely considered to be a promising research program that might lead to interesting results. It is also widely believed to necessitate new and extremely sophisticated pieces of mathematics in order to achieve its goal. The research program aims to prove the Ptime $\neq$ NPtime lower bound by showing that certain algebraic surfaces (representing the permanent and the discriminant, which are believed [28] to have different complexity if Ptime $\neq$ NPtime) cannot be embedded one into the other. Although this program has lead to interesting developments as far as pure mathematics is concerned, it has not enhanced our understanding of complexity lower bounds for the time being (actually, according to Mulmuley himself, such understanding will not be achieved in our lifetimes [12]). Recently, some negative results [15] have closed the easiest path towards it promised by GCT.

The GCT program was inspired, according to its creators, by a lower bound result obtained by Mulmuley [16]. Specifically, it was proved that the `maxflow` problem (deciding whether a certain quantity can flow from a source to a target in a weighted graph) is not solvable efficiently in a specific parallel model (the PRAM without bit operations). The `maxflow` problem is quite interesting as it is known to be in Ptime (by

[1]In the words of S. Aaronson and A. Wigderson [1], "We speculate that going beyond this limit [algebrization] will require fundamentally new methods."

reduction to linear programming, or the Ford-Fulkerson algorithm [11]), but there are no known efficient parallel algorithm solving it. This lower bound proof, despite being the main inspiration of the well-known GCT research program, remains seldom cited and has not led to variations applied to other problems. At first sight it relies a lot on algebraic geometric techniques and results, such as the Milnor-Thom theorem[2].

Dynamic Semantics The geometry of interaction program was proposed by Girard [13] shortly after the inception of linear logic. In opposition to traditional denotational semantics – e.g. domains –, the GOI program aims at giving an account of the proofs and programs which also interprets their dynamical features, i.e. cut-elimination/execution. This program is well-suited for tackling problems involving computational complexity, and indeed, geometry of interaction's first model was used to prove the optimality of Lamping's reduction in λ-calculus [14]. More recently, a series of characterisations of complexity classes were obtained using GOI techniques [4, 5, 2, 3].

Among the most recent and full-fledged embodiement of this program lie the second author's Interaction Graphs models [20, 22, 23, 24]. These models, in which proofs/programs are interpreted as *graphings* – generalisations of dynamical systems –, encompass all previous GOI models introduced by Girard [24]. In particular, Interaction Graphs allow for modelling quantitative features of programs/proofs [23].

2 Results

Computation models as graphings. The present work reports on the first investigations into how the interpretation of programs as graphings could lead to separation techniques, by rephrasing two well-known lower bound proofs, and strenghtening one of them.

The basic intuitions here can be summarised by the following slogan: "Computation, as a dynamical process, can be modelled as a dynamical system". Of course, the above affirmation cannot be true of all computational processes; for instance the traditional notion of dynamical system is deterministic. In practice, one works with a generalisation of dynamical systems named *graphings*; introduced as part of a family of models of linear logic, graphings model non-deterministic and probabilistic computation.

To do so, we consider that a computation model is given by a set of generators (that correspond to computation principles) and its actions on a space (representing the configuration space). So, in other words, we define a computation model as an action of a monoid (presented by its generators and relations) on a space $\alpha : M \curvearrowright \mathbf{X}$. A program in such a model of computation is then viewed as a graph, whose vertices are subspaces of the configuration space and edges are generators of the monoid: in this way, both the partiality of certain operations and branching is allowed.

Entropy We fix an action $\alpha : M \curvearrowright \mathbf{X}$ for the following discussion. One important aspect of the representation of abstract programs as graphings is that restrictions of graphings correspond to known notions from mathematics. In a very natural way, a deterministic α-graphing defines a partial dynamical system. Conversely, a partial dynamical system whose graph is contained in the *measured preorder* $\{(x,y) \in \mathbf{X}^2 \mid \exists m \in M, \alpha(m)(x) = y\}$ [21] can be associated to an α-graphing.

The study of deterministic models of computations can thus profit from the methods of the theory of dynamical systems. In particular, the methods employed in this paper relate to the classical notion of *topological entropy*. The topological entropy of a dynamical system is a value representing the average exponential growth rate of the number of orbit segments distinguishable with a finite (but arbitrarily fine) precision. The definition is based on the notion of open covers: for each finite open cover $\mathcal{C}$, one can compute the entropy of a map w.r.t. $\mathcal{C}$. As we are considering graphings and those correspond to partial maps, we explain how the techniques adapt to this more general setting and define the entropy $h(G, \mathcal{C})$ of a graphing G w.r.t. a cover $\mathcal{C}$.

[2]Let us here notice that, even though this is not mentionned by Mulmuley, the Milnor-Thom theorem was already used to prove lower bounds, c.f. papers by Dobkin and Lipton [10], Steele and Yao [27], Ben-Or [7], Cucker [9] and references therein.

The overall techniques related to entropy provide a much clearer picture of the techniques. In particular, the definition of *entropic co-trees* [18, Definition 37] are quite natural from this point of view and clarifies the methods employed by e.g. Ben-Or and Mulmuley.

Ben-Or's proof One lower bounds result related to Mulmuley's techniques is the bounds obtained by Steele and Yao [27] on *Algebraic Decision Trees.* Algebraic decision trees are defined as finite ternary trees describing a program deciding a subset of $\mathbf{R}^n$: each node verifies whether a chosen polynomial, say P, takes a positive, negative, or null value at the point considered. A d-th order algebraic decision tree is an algebraic decision tree in which all polynomials are of degree bounded by d.

In a very natural manner, an algebraic decision tree can be represented as an ι-graphings, when ι is the trivial action on the space $\mathbf{R}^n$. We use entropy to provide a bound on the number of connected components of subsets decided by ι-graphings. These bounds are obtained by combining a bound in terms of entropy and a variant of the Milnor-Thom theorem due to Ben-Or.

Theorem 1 (Theorem 28 [18])**.** *Let T be a d-th order algebraic decision tree deciding a subset $W \subseteq \mathbf{R}^n$. The number of connected components of W is bounded by $2^h d(2d-1)^{n+h-1}$, where h is the height of T.*

This result of Steele and Yao adapts in a straightforward manner to a notion of algebraic computation trees describing the construction of the polynomials to be tested by means of multiplications and additions of the coordinates. The authors remarked this result uses techniques quite similar to that of Mulmuley's lower bounds for the model of PRAMs *without bit operations.* It is also strongly similar to the techniques used by Cucker in proving that $\mathrm{NC}_{\mathbf{R}} \neq \mathrm{Ptime}_{\mathbf{R}}$ [9].

However, a refinement of Steele and Yao's method was quickly obtained by Ben-Or so as to obtain a similar result for an extended notion of algebraic computation trees allowing for computing divisions and taking square roots. Adapting Ben-Or techniques within the framework of graphings, we then apply this refined approach to Mulmuley's framework, leading to a strengthened lower bounds result.

The main result Using Ben-Or's technique to handle operations such as division and square root within PRAMs over integers, we improve over Mulmuley's proof. By considering that the length of an input is the minimal length of a binary word representing it, we get a realistic cost model for the PRAMs over integers, for which we can prove:

Theorem 63. *Let G be a* PRAM *without bit operations with $2^{O((\log N)^c)}$ processors, where N is the length of the inputs and c any positive integer. Then G does not decide* `maxflow` *in $O((\log N)^c)$ steps.*

If $\mathrm{NC}_{\mathbf{Z}}$ denotes the class of decision problems decided by a PRAM over integers in time and number of processors polylogarithmic in the length of the inputs, this proves:

$$\mathrm{NC}_{\mathbf{Z}} \neq \mathrm{Ptime}$$

Conclusion This work not only provides a strengthened lower bound results, but shows how the semantic techniques based on abstract models of computation and graphings can shed new light on some lower bound techniques. In particular, it establishes some relationship between the lower bounds and the notion of entropy which could potentially become deeper and provide new insights and finer techniques.

Showing that the interpretation of programs as graphings can translate, and even refine, such strong lower bounds results is also important from another perspective. Indeed, the techniques of Ben-Or and Mulmuley (as well as other results of e.g. Cucker [9], Yao [30]) seem at first sight restricted to algebraic models of computation due to their use of the Milnor-Thom theorem which holds only for real semi-algebraic sets. However, the second author's characterisations of Boolean complexity classes in terms of graphings acting on algebraic spaces [25] opens the possibility of using such algebraic methods to provide lower bounds for boolean models of computation.

References

[1] S. Aaronson and A. Wigderson. Algebrization: A new barrier in complexity theory. *ACM Trans. Comput. Theory*, 1(1):2:1–2:54, Feb. 2009.

[2] C. Aubert, M. Bagnol, P. Pistone, and T. Seiller. Logic programming and logarithmic space. In *APLAS*, 2014.

[3] C. Aubert, M. Bagnol, and T. Seiller. Unary resolution: Characterizing ptime. In *FOSSACS*, 2016.

[4] C. Aubert and T. Seiller. Characterizing co-nl by a group action. *Mathematical Structures in Computer Science*, 26:606–638, 2016.

[5] C. Aubert and T. Seiller. Logarithmic space and permutations. *Information and Computation*, 248:2–21, 2016.

[6] T. Baker, J. Gill, and R. Solovay. Relativizations of the p = np question. *SIAM Journal on Computing*, 4(4):431–442, 1975.

[7] M. Ben-Or. Lower bounds for algebraic computation trees. In *Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing*, STOC '83, pages 80–86, New York, NY, USA, 1983. ACM.

[8] S. Cook. The complexity of theorem-proving procedures. In *Proceedings of the 3rd ACM Symposium on Theory of Computing*, 1971.

[9] F. Cucker. $\mathbf{P_r} \neq \mathbf{NC_r}$. *Journal of Complexity*, 8(3):230 – 238, 1992.

[10] D. Dobkin and R. J. Lipton. Multidimensional searching problems. *SIAM Journal on Computing*, 5(2):181–186, 1976.

[11] L. R. Ford and D. R. Fulkerson. A simple algorithm for finding maximal network flows and an application to the hitchcock problem. *Canadian Journal of Mathematics*, pages 210–218, 1957.

[12] L. Fortnow. The status of the p versus np problem. *Commun. ACM*, 52(9):78–86, Sept. 2009.

[13] J.-Y. Girard. Towards a Geometry of Interaction. *Contemporary Mathematics*, 92:69–108, 1989.

[14] G. Gonthier, M. Abadi, and J.-J. Lévy. The geometry of optimal lambda reduction. In *Proceedings of the 19th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL '92, pages 15–26, New York, NY, USA, 1992. ACM.

[15] C. Ikenmeyer and G. Panova. Rectangular kronecker coefficients and plethysms in geometric complexity theory. *Advances in Mathematics*, 319:40 – 66, 2017.

[16] K. Mulmuley. Lower bounds in a parallel model without bit operations. *SIAM J. Comput.*, 28(4):1460–1509, 1999.

[17] K. D. Mulmuley. The gct program toward the p vs. np problem. *Commun. ACM*, 55(6):98–107, June 2012.

[18] L. Pellissier and T. Seiller. Prams over integers do not compute maxflow efficiently. Submitted, 2018.

[19] A. A. Razborov and S. Rudich. Natural proofs. *Journal of Computer and System Sciences*, 55(1):24 – 35, 1997.

[20] T. Seiller. Interaction graphs: Multiplicatives. *Annals of Pure and Applied Logic*, 163, 2012.

[21] T. Seiller. Towards a *Complexity-through-Realizability* theory. http://arxiv.org/pdf/1502.01257, 2015.

[22] T. Seiller. Interaction graphs: Additives. *Annals of Pure and Applied Logic*, 167, 2016.

[23] T. Seiller. Interaction graphs: Full linear logic. In *IEEE/ACM Logic in Computer Science (LICS)*, 2016.

[24] T. Seiller. Interaction graphs: Graphings. *Annals of Pure and Applied Logic*, 168(2):278–320, 2017.

[25] T. Seiller. Interaction graphs: Nondeterministic automata. *ACM Transaction in Computational Logic*, 19(3), 2018.

[26] T. Seiller, L. Pellissier, and U. Léchine. Unifying algebraic lower bounds, semantically. under revision for publication in Information and Computation, 2024.

[27] J. M. Steele and A. Yao. Lower bounds for algebraic decision trees. *Journal of Algorithms*, 3:1–8, 1982.

[28] L. G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8(2):189 – 201, 1979.

[29] R. Williams. Nonuniform acc circuit lower bounds. *J. ACM*, 61(1):2:1–2:32, Jan. 2014.

[30] A. C.-C. Yao. Decision tree complexity and betti numbers. *Journal of Computer and System Sciences*, 55(1):36 – 43, 1997.

Predicativism, Universality and Low-Complexity Computation

Amirhossein Akbar Tabatabai

Institute of Mathematics, Czech Academy of Sciences
amir.akbar@gmail.com

A function is primitive recursive iff it is representable by a map on the parameterized initial $F_{\mathbf{N}}$-algebra of any cartesian category (if it exists), where $F_{\mathbf{N}}(X) = 1 + X$. In this talk, following the philosophy of predicativism, we weaken the definition of a parameterized initial F-algebra to introduce a new notion called a *predicative F-scheme*, for any endofunctor F. Then, we show that the predicative $F_{\mathbf{N}}$-scheme (resp. predicative $F_{\mathbf{W}}$-scheme, where $F_{\mathbf{W}}(X) = 1 + X + X$) naturally captures the class of all linear space (resp. polynomial time) computable functions as its all and only representable functions. In the rest of this extended abstract, we will present the definitions of predicative F-schemes and representability to make the above points more formal.

First, we need to recall some basic definitions. Let $\mathcal{C}$ be a cartesian category (i.e., with all finite products), $F : \mathcal{C} \to \mathcal{C}$ be a functor and X be an object in $\mathcal{C}$. By an *F-algebra in $\mathcal{C}$ with parameters in X*, we mean the tuple $\mathbf{A} = (X, A, a)$, where $a : X \times F(A) \to A$ is a map in $\mathcal{C}$. The object A is called the *carrier* of $\mathbf{A}$ and is denoted by $|\mathbf{A}|$. When $X = 1$, an F-algebra with parameters in X is simply called an F-algebra. For any two F-algebras $\mathbf{A} = (X, A, a)$ and $\mathbf{B} = (X, B, b)$ in $\mathcal{C}$ with parameters in X, by an *F-homomorphism*, we mean a $\mathcal{C}$-map $f : A \to B$ such that the following diagram commutes:

$$\begin{array}{ccc} X \times F(A) & \xrightarrow{\ a\ } & A \\ \Big\downarrow{\scriptstyle id_X \times F(f)} & & \Big\downarrow{\scriptstyle f} \\ X \times F(B) & \xrightarrow[\ b\ ]{} & B \end{array}$$

It is clear that F-algebras in $\mathcal{C}$ with parameters in X together with F-homomorphisms form a category denoted by $\mathbf{Alg}^F_X(\mathcal{C})$. Moreover, the assignment $|-| : \mathbf{Alg}^F_X(\mathcal{C}) \to \mathcal{C}$ mapping an F-algebra with parameters in X to its carrier and an F-homomorphism to itself is a functor. Also, note that any $g : X \to Y$ in $\mathcal{C}$ induces a canonical functor $g^* : \mathbf{Alg}^F_Y(\mathcal{C}) \to \mathbf{Alg}^F_X(\mathcal{C})$.

Definition 1. Let $\mathcal{E}$ be a cartesian category, $\mathcal{D}$ be its (not necessarily full) cartesian subcategory, $i : \mathcal{D} \to \mathcal{E}$ be the inclusion functor preserving all finite products, and $F : \mathcal{E} \to \mathcal{E}$ be a functor whose restriction to $\mathcal{D}$ lands in $\mathcal{D}$ itself. An object I in $\mathcal{E}$ is called the *F-scheme of $\mathcal{D}$ in $\mathcal{E}$*, if for any $X \in \mathcal{D}$, the object $X \times I$ is the limit of the diagram $i|-| : \mathbf{Alg}^F_X(\mathcal{D}) \to \mathcal{E}$ via the cone $\langle r_{X,\mathbf{A}} \rangle_{\mathbf{A} \in \mathbf{Alg}^F_X(\mathcal{D})}$ and for any $\mathcal{D}$-map $f : X \to Y$ and any F-algebra $\mathbf{A}$ in $\mathcal{D}$ with parameters in Y, the following diagram commutes:

$$\begin{array}{ccc} I \times X & & \\ \Big\downarrow{\scriptstyle id_I \times f} & \searrow^{\,r_{X, f^*\mathbf{A}}} & \\ I \times Y & \xrightarrow[\ r_{Y,\mathbf{A}}\ ]{} & |\mathbf{A}| \end{array}$$

The F-scheme of $\mathcal{D}$ in $\mathcal{E}$ is meant to formalize the common *scheme* of all F-algebras of $\mathcal{D}$ (with parameters) inside the possibly greater category $\mathcal{E}$. Using the universality of the limit, one can easily show that there is a canonical F-algebra structure $a_I : F(I) \to I$ on I, whose composition with the projection provides an F-algebra structure on $I \times X$ with parameters in X. It is also easy to see that this algebraic structure makes all $r_{X,\mathbf{A}}$'s into F-homomorphisms.

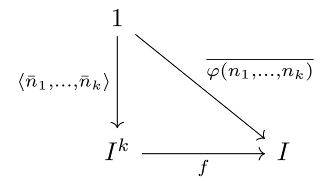
In the special case when $\mathcal{E} = \mathcal{D}$, the F-scheme of $\mathcal{D}$ in $\mathcal{D}$ is nothing but the initial F-algebra:

Theorem 2. *Let $\mathcal{D}$ be a finitely complete category. If I is the F-scheme of $\mathcal{D}$ in $\mathcal{D}$, then the F-algebra $a_I : F(I) \to I$ is the parameterized initial F-algebra in $\mathcal{D}$. Conversely, if $\mathbf{A}$ is the parameterized initial F-algebra in $\mathcal{D}$, then the object $|\mathbf{A}|$ together with its unique F-homomorphisms into the F-algebras of $\mathcal{D}$ (with parameters) is the F-scheme of $\mathcal{D}$ in $\mathcal{D}$.*

In the general situation when $\mathcal{E}$ is different from $\mathcal{D}$, we need to add an additional property, called the *approximability*, to gain a more well-behaved F-scheme. Roughly speaking, although the limit of the diagram $i|-| : \mathbf{Alg}^F_X(\mathcal{D}) \to \mathcal{E}$ may not belong to $\mathcal{D}$, we want it to be approximable by the objects inside the smaller category $\mathcal{D}$. More formally, the category $\mathbf{Alg}^F_X(\mathcal{D})$ is called approximable iff there is a directed family $\{\mathcal{S}_j\}_{j\in J}$ of classes of morphisms of $\mathcal{D}$ (not necessarily closed under composition) such that it covers the whole $Morph(\mathcal{D})$ and the restriction of $\mathbf{Alg}^F_X(\mathcal{D})$ to $\mathcal{S}_j$ has an initial element, for any $j \in J$. Unfortunately, the fully formal definition of approximability is beyond the scope of this short abstract. The reason is some subtleties in the definitions of the restriction, the initial element and the compatibility in the parameter object X, all because the $\mathcal{S}_j$'s are not necessarily closed under the composition.

Having approximability defined, the F-scheme of $\mathcal{D}$ in $\mathcal{E}$ is called *predicative* if $\mathbf{Alg}^F_X(\mathcal{D})$ is approximable.

Now, we turn to representability. Let $\mathbf{N} = (\mathbb{N}, s, 0)$ and $\mathbf{W} = (\mathbb{W}, s_0, s_1, \epsilon)$ be the usual algebras of natural numbers and binary strings, where $s(n) = n + 1$, $s_0(w) = w0$, $s_1(w) = w1$ and ϵ is the empty string. In the rest, let us assume that $\mathcal{D}$ and $\mathcal{E}$ are both cartesian and cocartesian categories, $i : \mathcal{D} \to \mathcal{E}$ preserves these structures and $F_{\mathbf{N}} : \mathcal{E} \to \mathcal{E}$ and $F_{\mathbf{W}} : \mathcal{E} \to \mathcal{E}$ be the functors defined by $F_{\mathbf{N}}(X) = 1 + X$ and $F_{\mathbf{W}}(X) = 1 + X + X$. It is possible to represent any element $n \in \mathbb{N}$ (resp. $w \in \mathbb{W}$) as a map in $Hom_{\mathcal{E}}(1, I)$, if I is the $F_{\mathbf{N}}$-scheme (resp. $F_{\mathbf{W}}$-scheme) of $\mathcal{D}$ in $\mathcal{E}$. Denote this canonical representation by $\bar{n}$ (resp. $\bar{w}$). Similarly, we say that an $\mathcal{E}$-map $f : I^k \to I$ represents a function $\varphi : \mathbb{N}^k \to \mathbb{N}$ if the following commutes:



for any $(n_1, \ldots, n_k) \in \mathbb{N}^k$. One can have a similar definition replacing $\mathbb{N}$ by $\mathbb{W}$. Now, we are finally ready to present our main result:

Theorem 3. (*i*) *A function $\varphi : \mathbb{N}^k \to \mathbb{N}$ is linear space computable iff it is representable as a map on the predicative $F_{\mathbf{N}}$-scheme of $\mathcal{D}$ in $\mathcal{E}$, for any $\mathcal{D}$ and $\mathcal{E}$.*

(*ii*) *A function $\varphi : \mathbb{W}^k \to \mathbb{W}$ is polynomial time computable iff it is representable as a map on the predicative $F_{\mathbf{W}}$-scheme of $\mathcal{D}$ in $\mathcal{E}$, for any $\mathcal{D}$ and $\mathcal{E}$.*

What is special about the 13^{th} Permutoassociahedron?

(Congruential functions as canonical isomorphisms)

Peter M. Hines – University of York

Although undecidability in mathematics is rightly attributed to the paradigm-shattering results of K. Gödel, exactly how widespread it is was brought home by the work of John Conway on *congruential functions*. These simple functions on the natural numbers — defined piece-wise linearly on modulo classes — are key to his demonstration of undecidability in elementary arithmetic [3, 4]. This was based on exhibiting computational universality of simple iterative problems on congruential functions via an encoding of Universal Register Machines, and then an appeal to the halting problem for Turing machines & equivalent systems.

The deceptively simple, but ultimately undecidable, problems considered by Conway were based on the orbits of natural numbers under some congruential function $\Gamma : \mathbb{N} \to \mathbb{N}$. We consider three distinct classes of functions :

pr-Fin All orbits are (provably) finite : $\left\|\{\Gamma^K(n)\}_{K\in\mathbb{N}}\right\| < \infty$ for all natural numbers $n \in \mathbb{N}$.

pr-Inf All orbits are (provably) either fixed points $\Gamma(n) = n$ or infinite $\left\|\{\Gamma^K(n)\}_{K\in\mathbb{N}}\right\| = \infty$, for all $n \in \mathbb{N}$.

un-Dec It is (in general) undecidable whether the orbit $\{\Gamma^K(n)\}_{K\in\mathbb{N}}$ of a given $n \in \mathbb{N}$ is finite or infinite.

Although Turing-completeness is well-studied categorically (usually via interpretations of logic or lambda calculus), Conway's congruential functions remain relatively little-studied for their categorical properties. The purpose of this talk is to demonstrate that a significant class – including many examples fundamental to numerous different areas of mathematics & theoretical computer science – are rightly seen as canonical coherence isomorphisms.

Our starting point is *exact covering systems* – a notion attributed to P. Erdös. These are pairwise-disjoint sets of modulo classes whose union is the whole of the natural numbers[1]. Number-theoretically, there are several methods of 'producing new covering systems from old'. We study the notion of "equally splitting a congruence class", which we demonstrate gives a somewhat disguised form of operadic composition on exact covering systems. Starting with the trivial exact covering (i.e. the set $\{\mathbb{N}\}$ itself), this procedure gives what are known as the "natural open covers", which we demonstrate form an operad isomorphic to the formal operad of rooted planar trees.

This then allows us to label associahedra with natural covering systems in a unique manner.

As a first step towards a categorical interpretation, we demonstrate how to build functors from exact covering systems, motivated by, but extending, constructions from J.-Y. Girard's Geometry of Interaction series of papers [5, 6, 7]. The functors derived from 'equally splitting' $\mathbb{N}$ itself form a "family of unbiased tensors" in the sense of [13] on a monoid of maps on the natural numbers. The natural covering systems then correspond to distinct bracketings of these unbiased tensors.

It is then almost trivial to demonstrate that the three classes of congruential functions determined by iterative properties (**pr-Fin**, **pr-Inf** and **un-Dec**) are closed under this family of unbiased tensors.

In a neat coincidence of notation between different fields, we derive natural isomorphisms from natural covering systems in an obvious manner. This gives a posetal functor category & so a notion of coherence for the above family of unbiased tensors. The components of the natural isomorphisms associated with this notion of coherence are not only easy to write down from their description as pairs of facets of some associahedron, but are precisely a distinguished set of congruential functions. A natural question is then how, or whether, we may characterise the iterative properties

[1] Algebraically, these may be thought of as disjoint basic open covers of the monoid $(\mathbb{N}, +)$, with respect to the profinite topology.

of thses canonical isomorphisms.

We study distinguished classes of these canonical isomorphisms, based on three aspects :

1. Their location on the associahedra,
2. Their previous appearance in other areas of mathematics & computer science,
3. Their behaviour under the iterative problems studied by Conway.

Our overall hypothesis is that these three aspects are closely related. Although this is a wide-ranging and indeed vaguely defined conjecture, we consider three classes of canonical isomorphisms for which this appears to hold true :

1. *(Non-commutative)* **Prime Factorisations**, *and* **pr-Fin**.
   A common view in the field of operads (e.g. [14]) is that the 'usual arithmetic' should be thought of as the commutative quotient of a non-commutative theory. We consider non-commutative analogues of multiplication and prime factorisation based on rooted planar trees, together with the corresponding natural isomorphisms / congruential functions between them. We demonstrate that these give congruential functions based on the radix-reversal permutations familiar from the Cooley-Tukey / Gauss theory of Fast Fourier Transforms. Not only are these canonical isomorphisms all in the class **pr-Fin** of functions with **provably finite** orbits, but they are also dual in a certain sense to those arising from Conway's `FRACTRAN` universal programming language.

2. **Vertices of Associahedra**, *and* **pr-Inf**.
   A distinguished class of mappings is given by those between vertices of associahedra. The connection between 1-skeleta of associahedra and MacLane's theory of coherence for associativity is of course well-known : from M. Kapranov, *"Given any n objects of a monoidal category, the associativity isomorphisms give a commuting diagram whose shape is the 1-skeleton of K_n"* – [11]. In our setting, we demonstrate that these give an isomorphic copy of Richard Thompson's group $\mathcal{F}$ as a group of congruential functions that have a very close connection with the class **pr-Inf** where orbits are either **infinite**, or **fixed points**.

3. **Boundary Maps**, *and* **un-Dec**.
   As we are considering unbiased tensors (i.e. one of each arity) with a notion of coherence, in the n^{th} associahedron $\mathcal{K}_n$ we may consider canonical isomorphisms between arbitrary facets – not simply those of the same dimension. Of particular interest are those between some facet, and the facets making up its boundary. We demonstrate that for every associahedron $\mathcal{K}_n$ (where $n \geq 3$), and every dimension $0 < x \leq n-2$, there exists an x-dimensional facet G and an $x-1$ dimension facet H on its boundary, where the canonical isomorphisms between G and H is precisely the congruential function described in [12] as claimed by Conway as his motivation for undecidability in arithmetic, and widely conjectured — following a probabilistic argument of Conway [4] — to be **undecidable** in any system powerful enough to express the problem, and hence in the **un-Dec** class.

If time permits, we will not only consider the categorical status of functions used in iterative arguments, but the categorical nature of the iterative process itself. This will be in terms of the *particle-style categorical trace* as a general model of iterative processes [10, 1, 8, 2, 9]. We will exhibit a particle-style trace on an inverse monoid of (partial) congruential functions equipped with a coherent family of unbiased tensors — providing a clear link between simple iterative problems, and the categorical structures used in models of computationally universal logics and lambda calculii.

References

[1] S. Abramsky. Retracing some paths in process algebra. In U. Montanari and V. Sassone, editors, *CONCUR '96: Concurrency Theory*, pages 1–17. Springer Berlin Heidelberg, 1996.

[2] S. Abramsky, E. Haghverdi, and P. Scott. Geometry of interaction and linear combinatory algebras. *Mathematical Structures in Computer Science*, 12 (5), 2002.

[3] John Conway. Unpredictable iterations. *Proc. 1972 Number Theory*, pages 49–52, 1972.

[4] John Conway. On unsettleable arithmetical problems. *The American Mathematical Monthly*, 120 (3):192–198, 2013.

[5] J.-Y. Girard. Geometry of interaction 1. In *Proceedings Logic Colloquium '88*, pages 221–260. North-Holland, 1988.

[6] J.-Y. Girard. Geometry of interaction 2: deadlock-free algorithms. In *Conference on Computer Logic*, volume 417 of *Lecture Notes in Computer Science*, pages 76–93. Springer, 1988.

[7] J.-Y. Girard. Geometry of interaction 3: Accommodating the additives. In *In: Advances in Linear Logic, LNS 222,CUP, 329–389*, pages 329–389. Cambridge University Press, 1995.

[8] P. Hines. *The algebra of self-similarity and its applications.* PhD thesis, University of Wales, Bangor, 1997.

[9] P. Hines. Machine semantics. *Theoretical Computer Science*, 409:1–23, 2008.

[10] A. Joyal, R. Street, and D. Verity. Traced monoidal categories. *Mathematical Proceedings of the Cambridge Philosophical Society*, 119(3):447–468, 1996.

[11] Mikhail M. Kapranov. The permutoassociahedron, mac lane's coherence theorem and asymptotic zones for the kz equation. *Journal of Pure and Applied Algebra*, 85(2):119–142, 1993.

[12] Jeffrey Lagarias. The $3x + 1$ problem and its generalizations. *The American Mathematical Monthly*, 92 (1):3–23, 1985.

[13] Tom Leinster. *Basic Category Theory.* Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2014.

[14] Jean-Louis Loday. Arithmetree. *Journal of Algebra*, 258(1):275–309, 2002. Special Issue in Celebration of Claudio Procesi's 60th Birthday.