Structure Meets Power Workshop

(Abstracts)

27–28 June 2021

https://www.cst.cam.ac.uk/conference/structure-meets-power-2021

Contents

Achim Blumensath: Compositionality and Logical Definability for Monads **3**

Pierre Pradic: Implicit automata in typed λ-calculi **4**

Martti Karvonen: Categorical composable cryptography **7**

Alexandre Goy: An Overview of Weak Distributive Laws **10**

John Power: Flexibility for a 2-monad **13**

Heba Aamer: Input–Output Disjointness for Forward Expressions in the Logic of Information Flows **16**

Alexandros Singh: Asymptotic Distribution of Parameters in Trivalent Maps and Linear Lambda Terms **19**

Nihil Shah: Bisimulation between hom sets and logics without counting: Extended abstract **23**

Peter Hines: On the combinatorics of Girard's exponential **26**

Thor Nielsen: Glued magic games self-test maximally entangled states **29**

Jan Hubička: Big Ramsey degrees using Ramsey theorem for trees with interesting levels **32**

Noam Zeilberger: Complexity of untyped normalization for subsystems of linear lambda calculus **35**

Steven Lindell: Traversal-invariant definability and Logarithmic-space computation **38**

Jakub Opršal: A categorical approach to constraint satisfaction problem **41**

Siddharth Bhaskar: Graph Traversals as Universal Constructions **44**

Glynn Winskel: On games on algebras **47**

Compositionality and Logical Definability for Monads

Achim Blumensath

In algorithmic model theory the so-called *composition method* is an established tool to study the expressive power of various logics. The general idea is to compute the theory of a large structure from the theories of several smaller ones into which it can be decomposed. We apply this method in the particular context of algebraic language theory. Recently there have been work to unify the various settings of algebraic language theory into a single uniform framework based on monads and Eilenberg-Moore algebras [2, 3, 1].

In this talk I present results from [1] that are organised around the following two questions.

(i) How does the composition method look like for a general monad and a general logic?

(ii) How can tools from algebraic language theory be used to answer definability questions in this setting?

The presentation will be mostly conceptual in nature with an emphasis on definitions over theorems.

The general idea is as follows. We consider an operation $\mathbb{M}$ that maps a set A to a class $\mathbb{M}A$ of A-labelled structures of some kind, and we study logics L talking about structures from $\mathbb{M}A$. In addition, we assume that the class $\mathbb{M}A$ is equipped with some sort of composition operation (turning $\mathbb{M}$ into a monad) and that our logic L is compatible with this operation. If $\mathbb{M}$ is sufficiently well-behaved, we can associate with every subclass $C \subseteq \mathbb{M}A$ a certain algebra $\mathfrak{S}$ in such a way that L-definability of C corresponds to certain algebraic properties of $\mathfrak{S}$.

References

[1] A. Blumensath, *Algebraic Language Theory for Eilenberg–Moore Algebras*, Logical Methods in Computer Science, 17 (2021), pp. 6:1–6:60.

[2] M. Bojańczyk, *Recognisable languages over monads.* unpublished note, arXiv:1502.04898v1.

[3] H. Urbat, J. Adámek, L.-T. Chen, and S. Milius, *Eilenberg theorems for free*, in 42nd International Symposium on Mathematical Foundations of Computer Science, MFCS 2017, August 21–25, 2017 – Aalborg, Denmark, vol. 83, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017, pp. 43:1–43:15.

Implicit automata in typed λ-calculi

Pierre Pradic
Department of Computer Science, University of Oxford, United Kingdom

This is joint work with Lê Thành Dũng Nguyễn (LIPN, Paris Nord).

The research that we present here started out by exploring connections between the languages (resp. functions) recognized by automata (resp. transducers, i.e., automata with output) and those definable by programs in certain typed λ-calculi. While we are relating logic and automata, much like programming language theory and proof theory are linked via the Curry–Howard correspondence, our work does not fit in the "logics as specification languages" paradigm, exemplified by the equivalence of finite-state automata and Monadic Second-Order Logic (MSO). One could sum up the difference by analogy with the two main approaches to machine-free complexity: *implicit computational complexity (ICC)* and *descriptive complexity*. Both aim to characterize complexity classes without reference to a machine model, but the methods of ICC have a more computational flavor.

programming paradigm	declarative	functional
complexity classes	Descriptive Complexity	Implicit Computational Complexity
automata theory	subsystems of MSO	**our work**

To our knowledge, very few works had previously looked at this kind of "type-theoretic" or "proof-theoretic" ICC for automata. Let us mention a few recent papers concerning transducers [5, 2] and multi-head automata [19, 10]. Most importantly, our starting point is a remarkable result dating back to 1996:

▶ **Theorem 1** (Hillebrand & Kanellakis [8, Theorem 3.4]). *A language $L \subseteq \Sigma^*$ can be defined in the* simply typed λ-calculus *by some* closed λ-term *of type $\mathtt{Str}_\Sigma[A] \to \mathtt{Bool}$ for some type A (that may depend on L) if and only if it is a* regular language.

Let us explain this statement. We consider a grammar of simple types with a single base type: $A, B ::= o \mid A \to B$, and use the *Church encodings* of booleans $\mathtt{Bool} = o \to o \to o$ and strings $\mathtt{Str}_\Sigma = (o \to o) \to \ldots \to (o \to o) \to o \to o$ with $|\Sigma|$ arguments of type $(o \to o)$ where Σ is a finite alphabet. For types A and B, we write $B[A]$ for the substitution $B\{o := A\}$ of every occurrence of o in B by A.

Although little-known, Hillebrand and Kanellakis's theorem is not surprising in retrospect: strong connections between Church encodings and automata (see e.g. [18, 11]) have been exploited, in particular in *higher-order model checking*.

Linear and non-commutative types While there exist some results in implicit complexity based on the simply typed λ-calculus (e.g. [8]), many works in that area have taken inspiration from *linear logic* to design more sophisticated type systems, starting with two characterizations of polynomial time [7, 6]. We followed the same idea to characterize two transduction classes in Intuitionistic Linear Logic with additives (ILL). From now on, $\mathtt{Str}_\Sigma$ denotes the linearized Church encoding $\mathtt{Str}_\Sigma = (o \multimap o) \to \ldots \to (o \multimap o) \to o \to o$.

▶ **Theorem 2** ([14]). *A function $\Gamma^* \to \Sigma^*$ is* regular (see e.g. [12]) (resp. comparison-free polyregular [13]) *if and only if it can be defined by a closed term of type $\mathtt{Str}_\Gamma[A] \multimap \mathtt{Str}_\Sigma$* (resp. $\mathtt{Str}_\Gamma[A] \to \mathtt{Str}_\Sigma$) *in ILL for some* purely linear *type A (that may depend on f).*

Here "purely linear" means that A does not contain any non-linear function arrow '$\to$'. We also provide a similar characterization of regular *tree* functions in [14]. What might be more

surprising is our additional use of *non-commutativity* (a function must use its arguments in the same order that they are given in) to characterize a subclass of regular languages.

▶ **Theorem 3** ([15]). *A language $L \subseteq \Sigma^*$ is* star-free *if and only if it can be defined by a closed term of type $\mathtt{Str}_\Sigma[A] \multimap \mathtt{Bool}$ in an affine variant of* Intuitionistic Non-Commutative Linear Logic [17] *for some* purely linear *type A (that may depend on L).*

Denotational semantics meets categorical automata theory The key conceptual insight in our proof of Theorem 2 is to relate the semantics of purely linear ILL in monoidal closed categories with the representation of transducers in Colcombet and Petrişan's categorical framework for automata [3]. More precisely, we show that a variant of *copyless* (i.e. affine) *streaming string transducers* [12, §2] – a machine model for regular functions – can be formulated as $\mathcal{C}$-automata where $\mathcal{C}$ is a Dialectica-like completion of a category of string-valued registers, so $\mathcal{C}$ is monoidal closed for the same reason as Dialectica categories [4].

The notion of monoidal closure has a relevance for automata theory that goes beyond Theorem 2; intuitively, this is due to the important role that function spaces often play in automata constructions. To illustrate that, in [14], we gave abstract generalizations of the arguments showing that copyless SSTs may be determinized and that the composition of two regular functions may be implemented by a copyless SST, in terms of internal homsets. Interestingly, it is not clear to us if there is a nice condition analogous to monoidal closure over classes of *transition monoids* allowing to carry out those generalized arguments without introducing automata over monoidal closed categories.

Some automata-theoretic consequences We were thus led to define the aforementioned *comparison-free polyregular functions* by considering expressible functions in linear logic. This class of function is a natural restriction of polyregular functions [2], a class of string transductions whose outputs are of size at most polynomial in the output. We studied that class in [13] from the point of view of automata theory. While all arguments are rather unsurprising, the connection with λ-calculus helped us realize that the class was closed under composition, and provides an alternative proof of this fact leveraging the material of [14].

Furthermore, we also took inspiration from Theorem 3 and from the planar geometry of interaction (GoI) semantics [1] of non-commutative linear logic to design a new machine model for star-free languages and aperiodic regular functions (see [12, §3] for the latter): *planar two-way automata/transducers* [16]. It was previously known that two-way automata could be expressed as automata over a GoI category (a reformulation of the results of [9] in the framework of [3]), and that two-way transducers compute regular functions [12, §2].

References

1 Samson Abramsky. Temperley–Lieb Algebra: From Knot Theory to Logic and Computation via Quantum Mechanics. In Goong Chen, Louis Kauffman, and Samuel Lomonaco, editors, *Mathematics of Quantum Computation and Quantum Technology*, volume 20074453, pages 515–558. Chapman and Hall/CRC, September 2007. doi:10.1201/9781584889007.ch15.

2 Mikołaj Bojańczyk. Polyregular functions, 2018. arXiv:1810.08760.

3 Thomas Colcombet and Daniela Petrişan. Automata Minimization: a Functorial Approach. *Logical Methods in Computer Science*, 16(1), March 2020. doi:10.23638/LMCS-16(1:32)2020.

4 Valeria C. V. de Paiva. The Dialectica categories. In John W. Gray and Andre Scedrov, editors, *Contemporary Mathematics*, volume 92, pages 47–62. American Mathematical Society, Providence, Rhode Island, 1989. doi:10.1090/conm/092/1003194.

5 Henry DeYoung and Frank Pfenning. Substructural proofs as automata. In Atsushi Igarashi, editor, *Programming Languages and Systems - 14th Asian Symposium, APLAS 2016, Hanoi, Vietnam, November 21-23, 2016, Proceedings*, volume 10017 of *Lecture Notes in Computer Science*, pages 3–22, 2016. doi:10.1007/978-3-319-47958-3_1.

6 Jean-Yves Girard. Light Linear Logic. *Information and Computation*, 143(2):175–204, June 1998. doi:10.1006/inco.1998.2700.

7 Jean-Yves Girard, Andre Scedrov, and Philip J. Scott. Bounded linear logic: a modular approach to polynomial-time computability. *Theoretical Computer Science*, 97(1):1–66, April 1992. doi:10.1016/0304-3975(92)90386-T.

8 Gerd G. Hillebrand and Paris C. Kanellakis. On the Expressive Power of Simply Typed and Let-Polymorphic Lambda Calculi. In *Proceedings of the 11th Annual IEEE Symposium on Logic in Computer Science*, pages 253–263. IEEE Computer Society, 1996. doi:10.1109/LICS.1996.561337.

9 Peter Hines. A categorical framework for finite state machines. *Mathematical Structures in Computer Science*, 13(3):451–480, 2003. doi:10.1017/S0960129503003931.

10 Denis Kuperberg, Laureline Pinault, and Damien Pous. Cyclic Proofs and Jumping Automata. In Arkadev Chattopadhyay and Paul Gastin, editors, *39th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2019)*, volume 150 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 45:1–45:14. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019. doi:10.4230/LIPIcs.FSTTCS.2019.45.

11 Paul-André Melliès. Higher-order parity automata. In *2017 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–12, Reykjavik, Iceland, June 2017. IEEE. doi:10.1109/LICS.2017.8005077.

12 Anca Muscholl and Gabriele Puppis. The Many Facets of String Transducers. In Rolf Niedermeier and Christophe Paul, editors, *36th International Symposium on Theoretical Aspects of Computer Science (STACS 2019)*, volume 126 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 2:1–2:21. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019. doi:10.4230/LIPIcs.STACS.2019.2.

13 Lê Thành Dũng Nguyễn, Camille Noûs, and Pierre Pradic. Comparison-free polyregular functions. Accepted to ICALP 2021, November 2020. URL: https://hal.archives-ouvertes.fr/hal-02986228.

14 Lê Thành Dũng Nguyễn, Camille Noûs, and Pierre Pradic. Implicit automata in typed λ-calculi II: streaming transducers vs categorical semantics, 2020. arXiv:2008.01050.

15 Lê Thành Dũng Nguyễn and Pierre Pradic. Implicit automata in typed λ-calculi I: aperiodicity in a non-commutative logic. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 168 of *LIPIcs*, pages 135:1–135:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ICALP.2020.135.

16 Lê Thành Dũng Nguyễn and Pierre Pradic. The planar geometry of first-order transductions, 2021. In preparation. Slides available at https://nguyentito.eu/2021-01-links.pdf.

17 Jeff Polakow and Frank Pfenning. Relating Natural Deduction and Sequent Calculus for Intuitionistic Non-Commutative Linear Logic. *Electronic Notes in Theoretical Computer Science*, 20:449–466, January 1999. doi:10.1016/S1571-0661(04)80088-4.

18 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. doi:10.1007/978-3-642-02261-6_5.

19 Thomas Seiller. Interaction Graphs: Non-Deterministic Automata. *ACM Transactions on Computational Logic*, 19(3):21:1–21:24, August 2018. doi:10.1145/3226594.

Categorical composable cryptography

Martti Karvonen (joint work with Anne Broadbent)

***Abstract*—This work formalize the simulation paradigm of cryptography in terms of category theory and shows that protocols secure against abstract attacks form a symmetric monoidal category, thus giving an abstract model of composable security definitions in cryptography. Our model is able to incorporate computational security, set-up assumptions and various attack models such as colluding or independently acting subsets of adversaries in a modular, flexible fashion. We are also able to use string diagrams to rederive no-go results concerning the limits of bipartite and tripartite cryptography, ruling out e.g. composable commitments and broadcasting. *Full version available at arXiv:2105.05949*.**

I. Introduction

Modern cryptographic protocols are complicated algorithmic entities, and their security analyses are often no simpler than the protocols themselves. Given this complexity, it would be highly desirable to be able to design protocols and reason about them compositionally, i.e. by breaking them down into smaller constituent parts. In particular, one would hope that combining protocols proven secure results in a secure protocol without need for further security proofs. However, this is not the case for stand-alone security notions that are common in cryptography. To illustrate such failures of composability, let us consider the history of quantum key distribution (QKD), as recounted in [24]: QKD was originally proposed in 80s [3]. The first security proofs against unbounded adversaries followed a decade later [4], [21], [27]. However, since composability was originally not a concern, it was later realized that the original security definitions did not provide a good enough level of security [15]—they didn't guarantee security if the keys were to be actually used, since even a partial leak of the key would compromise the rest. The story ends on a positive note, as eventually a new security criterion was proposed, together with stronger proofs [2], [26].

In arXiv:2105.05949, we initiate a categorical study of composable security definitions in cryptography. In the viewpoint developed there one thinks of cryptography as a resource theory: cryptographic functionalities (e.g. secure communication channels) are viewed as resources and cryptographic protocols let one transform some starting resources to others. For instance, one can view the one-time-pad as a protocol that transforms an authenticated channel and a shared secret key into a secure channel. For a given protocol, one can then study whether it is secure against some (set of) attack model(s), and protocols secure against a fixed set of models can always be composed sequentially and in parallel.

This is in fact the viewpoint taken in constructive cryptography [19], which also develops the one-time-pad example above in more detail. However [19] does not make a formal connection to resource theories as usually understood, whether as in quantum physics [9], [14], or more generally as defined in order theoretic [11] or categorical [10] terms. Instead, constructive cryptography is usually combined with abstract cryptography [20] which is formalized in terms of a novel algebraic theory of systems [18].

Our work can be seen as a particular formalization of the ideas behind constructive cryptography, or alternatively as giving a categorical account of the real-world-ideal-world paradigm (also known as the simulation paradigm [12]), which underlies more concrete frameworks for composable security, such as universally composable cryptography [7] and others [1], [13], [16], [17], [22].

Our long-term goal is to enable cryptographers to reason about composable security at the same level of formality as stand-alone security, *without having to fix all the details of a machine model nor having to master category theory*. Indeed, our current results already let one define multipartite protocols and security against arbitrary subsets of malicious adversaries *in any symmetric monoidal category* $\mathbf{C}$. Thus, as long as one's model of interactive computation results in a symmetric monoidal category, or more informally, one is willing to use pictures such as those appearing in the sequel, to depict connections between computational processes without further specifying the order in which the picture was drawn, one can use the simulation paradigm to reason about multipartite security against malicious participants composably—and specifying finer details of the computational model is only needed to the extent that it affects the validity of one's argument. Moreover, as our attack models and composition theorems are fairly general, we hope that more refined models of adversaries can be incorporated.

We now highlight some of the contributions of arXiv:2105.05949:

- We show how to adapt resource theories as categorically formulated [10] in order to reason abstractly about *secure* transformations between resources. This is done by formalizing the simulation paradigm in terms of an abstract attack model, designed to be general enough to capture standard attack models of interest (and more) while still structured enough to guarantee composability. We show that for any fixed set of attack models, the class of protocols secure against each of them results in a symmetric monoidal category.
- We adapt this framework to model *computational security* in two ways: either by replacing equations with an equivalence relation, abstracting the idea of computational indistinguishability, or by working with a notion of distance. In the case of a distance, one can then either explicitly bound the distance between desired and

actually achieved behavior, or work with sequences of protocols that converge to the target in the limit: the former models working in the finite-key regimen [28] and the latter models the kinds of asymptotic security and complexity statements that are common in cryptography.
- Finally, we apply the framework developed to study bipartite and tripartite cryptography. We reprove the no-go-theorems of [18], [20], [25] concerning two-party commitments (and three-party broadcasting) in this setting, and reinterpret them as limits on what can be achieved securely in any compact closed category (symmetric monoidal category). The key steps of the proof are done graphically, thus opening the door for cryptographers to use such pictorial representations as rigorous tools rather than merely as illustrations.

II. Sketching the approach

As shown in [10], many familiar resource theories arise in a uniform fashion: starting from an SMC (symmetric monoidal category) $\mathbf{C}$ of processes equipped with a wide sub-SMC $\mathbf{C}_F$ whose morphisms represent "free" processes, they build several resource theories (=SMCs). Perhaps the most important of these constructions is the resource theory of states: given $\mathbf{C}_F \hookrightarrow \mathbf{C}$, the corresponding resource theory of states can be explicitly constructed by taking the objects of this resource theory to be states of $\mathbf{C}$, i.e. maps $r\colon I \to A$ for some A, and maps $r \to s$ are maps $f\colon A \to B$ in $\mathbf{C}_F$ such that $fr = s$

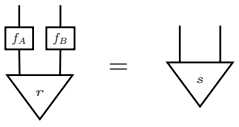
Our first observation is that there is no reason to restrict to inclusions $\mathbf{C}_F \hookrightarrow \mathbf{C}$ in order to construct a resource theory of states. This is because the resulting category is the category of elements of the composite $\mathbf{C}_F \to \mathbf{C} \xrightarrow{\hom(I,-)} \mathbf{Set}$. As this is a (lax) symmetric monoidal functor, the resulting category is automatically symmetric monoidal as observed in [23]. Thus this construction goes through for any symmetric (lax) monoidal functors $\mathbf{D} \xrightarrow{F} \mathbf{C} \xrightarrow{R} \mathbf{Set}$. Here we may think of F as interpreting free processes into an ambient category of all processes, and $R\colon \mathbf{C} \to \mathbf{Set}$ as an operation that gives for each object A of $\mathbf{C}$ the set $R(A)$ of resources of type A.

Consider now the resource theory induced by $\mathbf{C}^n \xrightarrow{\bigotimes} \mathbf{C} \xrightarrow{\hom(I,-)} \mathbf{Set}$, where we write $\bigotimes$ for the n-fold monoidal product[1]. The resulting resource theory has a natural interpretation in terms of n agents trying to transform resources to others: an object of this resource theory corresponds to a pair $((A_i)_{i=1}^n, r\colon I \to \bigotimes A_i)$, and can be thought of as a state where the i-th agent has access to a port of type A_i. A morphism $\bar{f} = (f_1, \dots f_n)\colon ((A_i)_{i=1}^n, r) \to ((B_i)_{i=1}^n, s)$ between such resources then amounts to a protocol that prescribes, for each agent i a process f_i that they should perform so that $(\bigotimes_i f_i)r = s$ In this resource theory, all of the agents are equally powerful and can perform all processes allowed by $\mathbf{C}$, and this might be unrealistic: first of all, $\mathbf{C}$ might include computational processes that are too powerful/expensive for us to use in our cryptographic protocols. Moreover, having agents with different computational powers is important to model e.g. blind quantum computing [5] where a client with access only to limited, if any, quantum computation tries to securely delegate computations to a server with a powerful quantum computer. This limitation is easily remedied: we could take the i-th agent to be able to implement computations in some sub-SMC $\mathbf{C}_i$ of $\mathbf{C}$, and then consider $\prod_{i=1}^n \mathbf{C}_i \to \mathbf{C}$.

A more serious limitation is that such transformations have no security guarantees—they only work if each agent performs f_i as prescribed by the protocol. In order for a protocol $\bar{f} = (f_1, \dots, f_n)\colon ((A_i)_{i=1}^n, r) \to ((B_i)_{i=1}^n, s)$ to be secure, we should have some guarantees what happens if, as a result of *an attack* on the protocol, something else than $(f_1, \dots, f_n)$ happens. For instance, some subset of the parties might deviate from the protocol and do something else instead. In the simulation paradigm, security is then defined by saying that, anything that could happen when running the real protocol, i.e., $\bar{f}$ with r, could also happen in the ideal world, i.e. with s. A given protocol might be secure against some kinds of attacks and insecure against others, so in arXiv:2105.05949 we define security against an abstract attack model.

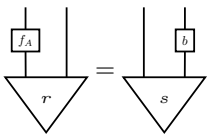
Instead of general results, we discuss the resulting definitions in a special case. We assume $n = 2$, and focus on perfect security against either party behaving maliciously. In this case we may define a protocol $\bar{f} = (f_A, f_B)$ to be a secure protool $r \to s$ if

(i)



(Correctness)

(ii) There exists a morphism b such that



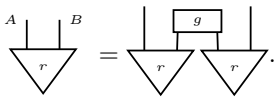
(Security against Bob)

(iii) Similarly for Alice.

Composable security is a stronger constraint than stand-alone security, and many cryptographic functionalities are known to be impossible to achieve "in the plain model", i.e. without set-up assumptions. A case in point is bit commitment, which was shown to be impossible in the UC-framework in [8]. This result was later generalized in [25] to show that any two-party functionality that can be realized in the plain UC-framework is "splittable". We present a categorical proof of this result, which promotes the pictures "illustrating the proof" in [25] into a full proof — the main difference is that in [25] the pictures explicitly keep track of an environment trying to distinguish between different functionalities, whereas we prove our result in the case of perfect security and then deduce the asymptotic claim. We now assume that $\mathbf{C}$, our ambient category of interactive computations is compact closed.

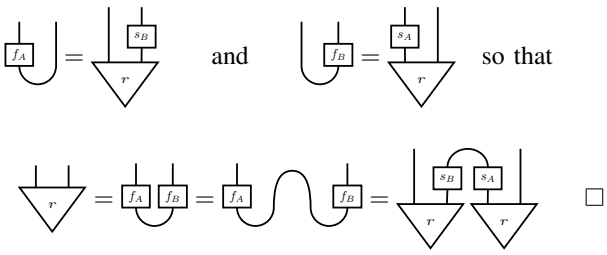
[1] As $\mathbf{C}$ is symmetric, the functor $\bigotimes$ is strong monoidal.

Theorem II.1. *For Alice and Bob (one of whom might cheat), if a bipartite functionality r can be securely realized from a communication channel between them, i.e. from $\cup$, then there exists a g such that*



(∗)

Proof. Assume a protocol (f_A, f_B) achieving this. Security constraints against each party give us



□

Corollary II.2. *Given a compact closed* **C** *modeling computation in which wires model communication channels, (composable) bit commitment and oblivious transfer are impossible in that model without setup, even asymptotically in terms of distinguisher advantage.*

Proof. If r represents bit commitment from Alice to Bob, it does not satisfy the equation required by Theorem II.1 for any f, and the two sides of $(*)$ can be distinguished efficiently with at least probability $1/2$. Indeed, take any f and let us compare the two sides of $(*)$: if the distinguisher commits to a random bit b, then Bob gets a notification of this on the left hand-side, so that f has to commit to a bit on the right side of $(*)$ to avoid being distinguished from the left side. But this bit coincides with b with probability at most $1/2$, so that the difference becomes apparent at the reveal stage. The case of OT is similar. □

REFERENCES

[1] Michael Backes, Birgit Pfitzmann, and Michael Waidner. The reactive simulatability (rsim) framework for asynchronous systems. *Information and Computation*, 205(12):1685–1720, 2007. doi:10.1016/j.ic.2007.05.002.

[2] Michael Ben-Or, Michał Horodecki, Debbie W Leung, Dominic Mayers, and Jonathan Oppenheim. The universal composable security of quantum key distribution. In *2nd Theory of Cryptography Conference—TCC 2005*, pages 386–406, 2005. doi:10.1007/978-3-540-30576-7_21.

[3] Charles H. Bennett and Gilles Brassard. Quantum cryptography: Public key distribution and coin tossing. In *International Conference on Computers, Systems and Signal Processing*, pages 175–179, 1984.

[4] Eli Biham, Michel Boyer, P. Oscar Boykin, Tal Mor, and Vwani Roychowdhury. A proof of the security of quantum key distribution (extended abstract). In *32nd Annual ACM Symposium on Theory of Computing—STOC 2000*, pages 715 – 724, 2000. doi:10.1145/335305.335406.

[5] Anne Broadbent, Joseph Fitzsimons, and Elham Kashefi. Universal blind quantum computation. In *50th Annual Symposium on Foundations of Computer Science—FOCS 2009*, pages 517–526, 2009. doi:10.1109/FOCS.2009.36.

[6] Anne Broadbent and Martti Karvonen. Categorical composable cryptography, 2021. arXiv:2105.05949.

[7] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *42nd Annual Symposium on Foundations of Computer Science—FOCS 2001*, pages 136–145, 2001. doi:10.1109/SFCS.2001.959888.

[8] Ran Canetti and Marc Fischlin. Universally composable commitments. In *Advances in cryptology—CRYPTO 2001*, pages 19–40. Springer, 2001. doi:10.1007/3-540-44647-8_2.

[9] Eric Chitambar and Gilad Gour. Quantum resource theories. *Reviews of Modern Physics*, 91(2):025001, 2019. doi:10.1103/revmodphys.91.025001.

[10] Bob Coecke, Tobias Fritz, and Robert W Spekkens. A mathematical theory of resources. *Information and Computation*, 250:59–86, 2016. doi:10.1016/j.ic.2016.02.008.

[11] Tobias Fritz. Resource convertibility and ordered commutative monoids. *Mathematical Structures in Computer Science*, 27(6):850–938, 2015. doi:10.1017/s0960129515000444.

[12] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984. doi:10.1016/0022-0000(84)90070-9.

[13] Dennis Hofheinz and Victor Shoup. GNUC: A new universal composability framework. *Journal of Cryptology*, 28(3):423–508, 2015. doi:10.1007/s00145-013-9160-y.

[14] Michal Horodecki and Jonathan Oppenheim. (quantumness in the context of) resource theories. *International Journal of Modern Physics B*, 27(01n03):1345019, 2013. doi:10.1142/s0217979213450197.

[15] Robert König, Renato Renner, Andor Bariska, and Ueli Maurer. Small accessible quantum information does not imply security. *Physical Review Letters*, 98(14):140502, 2007. doi:10.1103/PhysRevLett.98.140502.

[16] Ralf Küsters, Max Tuengerthal, and Daniel Rausch. The IITM model: a simple and expressive model for universal composability. *Journal of Cryptology*, 33(4):1461–1584, 2020. doi:10.1007/s00145-020-09352-1.

[17] Kevin Liao, Matthew A. Hammer, and Andrew Miller. ILC: a calculus for composable, computational cryptography. In *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation*, pages 640–654. ACM, June 2019. doi:10.1145/3314221.3314607.

[18] Christian Matt, Ueli Maurer, Christopher Portmann, Renato Renner, and Björn Tackmann. Toward an algebraic theory of systems. *Theoretical Computer Science*, 747:1–25, 2018. doi:10.1016/j.tcs.2018.06.001.

[19] Ueli Maurer. Constructive cryptography–a new paradigm for security definitions and proofs. In *Joint Workshop on Theory of Security and Applications—TOSCA 2011*, pages 33–56, 2011. doi:10.1007/978-3-642-27375-9_3.

[20] Ueli Maurer and Renato Renner. Abstract cryptography. In *Innovations in Computer Science—ICS 2011*, 2011.

[21] Dominic Mayers. The trouble with quantum bit commitment, 1996. URL: http://arxiv.org/abs/quant-ph/9603015, arXiv:9603015.

[22] Daniele Micciancio and Stefano Tessaro. An equational approach to secure multi-party computation. In *4th Conference on Innovations in Theoretical Computer Science—ITCS 2013*, pages 355–372, 2013. doi:10.1145/2422436.2422478.

[23] Joe Moeller and Christina Vasilakopoulou. Monoidal Grothendieck construction. *Theory and Applications of Categories*, 35(31):1159–1207, 2020.

[24] Christopher Portmann and Renato Renner. Cryptographic security of quantum key distribution, 2014. arXiv:1409.3525.

[25] Manoj Prabhakaran and Mike Rosulek. Cryptographic complexity of multi-party computation problems: Classifications and separations. In *Advances in Cryptology—CRYPTO 2008*, pages 262–279, 2008. doi:10.1007/978-3-540-85174-5_15.

[26] Renato Renner. Security of quantum key distribution. *International Journal of Quantum Information*, 06(01):1–127, 2005. arXiv:0512258v2, doi:10.1142/S0219749908003256.

[27] Peter W. Shor and John Preskill. Simple proof of security of the BB84 quantum key distribution protocol. *Physical Review Letters*, 85(2):441–444, 2000. doi:10.1103/physrevlett.85.441.

[28] Marco Tomamichel, Charles Ci Wen Lim, Nicolas Gisin, and Renato Renner. Tight finite-key analysis for quantum cryptography. *Nature Communications*, 3:634, 2012. doi:10.1038/ncomms1631.

An Overview of Weak Distributive Laws

Alexandre Goy

Université Paris-Saclay, CentraleSupélec, MICS
Gif-sur-Yvette, France
alexandre.goy@centralesupelec.fr

Abstract

Distributive laws have proved useful to combine effects of two monads. In frequent cases though, there is no way of defining a distributive law between a pair of specific monads. When it feels like there *almost* exists one, we can use a weaker form of distributive law instead. This talks gives an overview of how weak distributive laws can be applied in computer science. Part of the results have been published in [8] or will be published in [10]. Some results, e.g. coweak distributive laws, have not been published. This research is joint work with Daniela Petrişan and, as regards toposes, with Marc Aiguier.

A *monad* (T, η^T, μ^T) on a category C is a functor $T : C \to C$ along with two natural transformations $\eta^T : 1 \to T$ and $\mu^T : TT \to T$ such that $\mu^T \circ T\eta^T = \mu^T \circ \eta^T T = 1$ and $\mu^T \circ T\mu^T = \mu^T \circ \mu^T T$. Monads have been originally imported in computer science to model computational effects [12, 13], and nowadays constitute a must-have concept in one's categorical toolbox. A famous issue in the theory of monads is its non-compositionality. Given two monads (S, η^S, μ^S) and (T, η^T, μ^T), there is no guarantee that the functor ST has a monad structure. However, in the presence of a *distributive law*, i.e. a natural transformation $\lambda : TS \to ST$ satisfying the four axioms of Table 1, there is a monad $(ST, \eta^S\eta^T, \mu^S\mu^T \circ S\lambda T)$ [1]. Notably, each distributive law $TS \to ST$ is equivalently a lifting of S to the category of T-algebras, and an extension of T to the Kleisli category of S. Again, a famous issue in the theory of distributive laws is that in many interesting cases, there simply is no distributive law. For example, in Set, the powerset monad P and the distribution monad D interact rather badly: there is no distributive law $PP \to PP$ [11], $DP \to PD$ [15], $PD \to DP$ [15, 16] nor $DD \to DD$ [16]. This is not an isolated behaviour, and one can find many other concrete examples in Zwart's thesis [16].

In the last few years, some authors have been working to weakening the notion of distributive law [2, 14, 3, 7]. We retain the definition of Garner: a *weak distributive law* is a natural transformation $\lambda : TS \to ST$ such that all of the axioms in Table 1, except the (η^T) one, hold. Under the mild assumption that idempotents split in C, any weak distributive law is equivalently a (weak form of) lifting of S to the category of T-algebras, and a (weak form of) extension of T to the Kleisli category of S. One can also obtain – out of the weak lifting – a tweaked composite monad involving features of both S and T, but whose functor is not required to be ST.

Table 1: Axioms of distributive laws $TS \to ST$

name	axiom	plain	weak	coweak
(η^S)	$\lambda \circ T\eta^S = \eta^S T$	✓	✓	×
(η^T)	$\lambda \circ \eta^T S = S\eta^T$	✓	×	✓
(μ^S)	$\lambda \circ T\mu^S = \mu^S T \circ S\lambda \circ \lambda S$	✓	✓	✓
(μ^T)	$\lambda \circ \mu^T S = S\mu^T \circ \lambda T \circ T\lambda$	✓	✓	✓

It turns out that there are lots of weak distributive laws. For example, any monad morphism $\sigma : S \to T$ gives rise to a weak distributive law $\eta^S \mu^T \circ T\sigma : TS \to ST$ – but the behaviour of such laws is largely irrelevant. In Set and with S being the powerset monad P, a powerful way of distinguishing useful weak distributive laws is to use the following result.

Theorem 1 ([7]). *There is a (unique) monotone weak distributive law $TP \to PT$ if and only if T preserves weak pullbacks and μ^T naturality squares are weak pullbacks.*

This result can be adapted to regular categories that comprise a good notion of powerset monad. A number of weak distributive laws can be obtained via Theorem 1 or its variants.

- There is weak distributive law $\beta P \to P\beta$ [7], where β is the ultrafilter monad. The corresponding weak lifting is the Vietoris monad V on the category KHaus of compact Hausdorff spaces and continuous functions.
- There is a weak distributive law $DP \to PD$ [8]. The corresponding weak lifting is the convex powerset monad on the category of convex algebras.
- Under appropriate conditions on the semiring s, there is a weak distributive law $SP \to PS$ [4], where here S denotes the s-left-semimodule monad.
- There is a weak distributive law $PP \to PP$ [7, 9].
- More generally, any topos has a powerset P and a weak distributive law $PP \to PP$ [10].
- With the Vietoris monad V playing the role of a powerset in KHaus, there is also a similar distributive law $VV \to VV$ [10].

Interestingly enough, many constructions that are usually performed using distributive laws can still be carried out with respect to weak ones. For example, one can obtain a composite algebraic theory of the tweaked monad from algebraic theories of the basic monads [7], or one can build iterated weak distributive laws in the style of Cheng [6]. One can define monad-functor weak distributive laws $TF \to FT$ by forgetting the (η^S) and (μ^S) axioms, thereby recovering the generalized powerset construction for FT-coalgebras [8, 9], as well as compatibility of up-to techniques [9]. For example, the $DP \to PD$ weak distributive law entails that bisimilarity up-to convex hull for probabilistic automata (see [5]) is sound.

Many interesting questions remain open. To begin with, further instances of weak distributive laws remain to be found. For example, the continuous equivalent of the distribution monad D on Set is the Radon monad R on KHaus: can the law $DP \to PD$ be generalized to a law $RV \to VR$, using a variant of Theorem 1? Answering this question is actually equivalent to proving four properties about R, out of which we only know two at the present time. Next, the scope of Theorem 1 is very restrictive, so how can we identify interesting weak distributive laws when S is not "powerset-like"?

Another interesting line of research is to look at the dual of weak distributive laws, hereby called *coweak* distributive laws. Such laws require the (η^T) axiom but do not require the (η^S) axiom (see Table 1). It turns out that under the assumption that idempotents split in the Kleisli category of S $(*)$, a coweak distributive law is equivalently a (coweak form of) lifting of S to T-algebras and a (coweak form of) extension of T to the Kleisli category of S. Additionally, one can build – out of the coweak extension – a tweaked composite monad. Note that even if idempotents split in C, they may not split in Kleisli of S (take e.g. $C = \mathsf{Set}$ and $S = P$), so that the condition $(*)$ can not be considered as mild anymore. Concrete non-trivial examples of coweak distributive laws are still to be found.

2

References

[1] Jon Beck. Distributive laws. In B. Eckmann, editor, *Seminar on Triples and Categorical Homology Theory*, pages 119–140, Berlin, Heidelberg, 1969. Springer Berlin Heidelberg.

[2] Gabriella Böhm. The weak theory of monads. *Advances in Mathematics*, 225(1):1–32, 2010.

[3] Gabriella Böhm, Stephen Lack, and Ross Street. On the 2-categories of weak distributive laws. *Communications in Algebra*, 39(12):4567–4583, 2011.

[4] Filippo Bonchi and Alessio Santamaria. Combining semilattices and semimodules, 2021.

[5] Filippo Bonchi, Alexandra Silva, and Ana Sokolova. The Power of Convex Algebras. In Roland Meyer and Uwe Nestmann, editors, *28th International Conference on Concurrency Theory (CONCUR 2017)*, volume 85 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 23:1–23:18, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[6] Eugenia Cheng. Iterated distributive laws. *Mathematical Proceedings of the Cambridge Philosophical Society*, 150(3):459–487, 2011.

[7] Richard Garner. The Vietoris monad and weak distributive laws. *Applied Categorical Structures*, 28(2):339–354, 4 2020.

[8] Alexandre Goy and Daniela Petrişan. Combining probabilistic and non-deterministic choice via weak distributive laws. In *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '20, page 454–464. Association for Computing Machinery, 2020.

[9] Alexandre Goy and Daniela Petrişan. Combining weak distributive laws: Application to up-to techniques. *CoRR*, abs/2010.00811, 2020.

[10] Alexandre Goy, Daniela Petrişan, and Marc Aiguier. Powerset-like monads weakly distribute over themselves in toposes and compact Hausdorff spaces. In *Proceedings of the 48th International Colloquium on Automata, Languages, and Programming*, ICALP '21. Association for Computing Machinery, 2021.

[11] Bartek Klin and Julian Salamanca. Iterated covariant powerset is not a monad. *Electronic Notes in Theoretical Computer Science*, 341:261–276, 2018. Proceedings of the Thirty-Fourth Conference on the Mathematical Foundations of Programming Semantics (MFPS XXXIV).

[12] Eugenio Moggi. Notions of computation and monads. *Information and Computation*, 93(1):55–92, 1991. Selections from 1989 IEEE Symposium on Logic in Computer Science.

[13] Gordon Plotkin and John Power. Notions of computation determine monads. In Mogens Nielsen and Uffe Engberg, editors, *Foundations of Software Science and Computation Structures*, pages 342–356, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[14] Ross Street. Weak distributive laws. *Theory and Applications of Categories [electronic only]*, 22:313–320, 2009.

[15] Daniele Varacca. *Probability, nondeterminism and concurrency: two denotational models for probabilistic computation*. BRICS, 2003.

[16] Maaike Zwart. *On the non-compositionality of monads via distributive laws*. PhD thesis, Oxford, 2020.

Flexibility for a 2-Monad

John Power
Department of Computing, Macquarie University, NSW 2009
Australia

A central theme of category theory from its inception has been restriction of the expressiveness of set theory. For instance, the set theoretic product of sets X and Y is a specific set, with specified elements determined by those of X and Y, whereas the category theoretic product of X and Y is only determined up to coherent isomorphism. So the category theoretic definition does not specify a specific set but rather only determines a class of sets that satisfy the key mathematical feature of a product. Such contrast also applies to the coproduct of sets and to their exponential.

This is part of a broader phenomenon. In general, category theory focuses almost exclusively upon structures on sets that are invariant with respect to isomorphism, whether that be isomorphism of algebraic structure or homeomorphism of topological structure. In this specific aspect, it better reflects the ordinary discourse of mathematics than set theory does. It involves a subtle mosaic of ideas. One such idea is exemplified by the following: given a group G and a bijection between the underlying set of G and a set X, the group structure of G transports along the bijection uniquely to give a group structure on X. Here, we investigate the extension of that fact from sets with structure to categories with structure.

When one extends from sets with structure to categories with structure, the situation becomes more complex. In particular, there are convenient structures on categories that do not transport along equivalence. For instance, given a small category C, the category $End(C)$ of endofunctors on C has a strict monoidal structure given by composition of functors; strict monoidal structure does not, in general, transport along equivalence; but the strictness of that monoidal structure is helpful, simplifying calculations.

However, it remains the case that, with a few exceptions such as the above, the structures of primary interest on categories, such as products, coproducts, monoidal structure, exponentials, or subobject classifiers, do transport along equivalence. So, in a conceptual setting that includes such examples,

we seek a restriction on the expressiveness of categories with structure to those that allow transport of structure along equivalence. For simplicity of exposition, I shall restrict attention to covariant structure, thus including limits, colimits and monoidal structure with assorted additional structure such as a symmetry, but excluding exponentials or subobject classifiers.

Probably the most prominent systematic account of categories with structure to date has been given by finitary 2-monads T on Cat [1], equivalently by Lawvere 2-theories. Perhaps counter-intuitively in light of the above, as explained in detail in [1], the focus is on strict T-algebras. For instance, there is a 2-monad T_{fp} for which the strict algebras are small categories with assigned finite products; there is one, T_{sm}, for which the strict algebras are small strict monoidal categories; and there is one, T_m, whose strict algebras are small monoidal categories.

There is more than one kind of map of strict T-algebras. *Strict* maps of T-algebras instantiate to functors that strictly preserve assigned finite products, monoidal structure, etcetera. *Pseudo* maps of T-algebras instantiate to functors that preserve structure in the usual sense, i.e., up to coherent isomorphism. Strict T-algebras and pseudo-maps, with evident 2-cells, form a 2-category $T\text{-}Alg$ that, together with its lax variant, form the central object of study of [1], with $T\text{-}Alg_s$ denoting the sub-2-category determined by strict maps. The central technical theorem of [1] asserts that, for an arbitrary finitary 2-monad T on an arbitrary locally finitely presentable 2-category $\mathcal{K}$, the inclusion $J : T\text{-}Alg_s \longrightarrow T\text{-}Alg$ has a left adjoint, denoted by $(-)'$.

One can also routinely define *pseudo*-algebras, yielding the 2-category $Ps\text{-}T\text{-}Alg$ of pseudo-T-algebras and pseudo-maps, duly containing $T\text{-}Alg$ as a full subcategory.

It is routine to show that, in general, for any 2-monad T on any 2-category $\mathcal{K}$, given a strict T-algebra (C, c) and any 0-cell D together with an equivalence $e : C \longrightarrow D$, the strict T-structure of C transports along e to a pseudo-T-structure on D, yielding an equivalence in the 2-category $Ps\text{-}T\text{-}Alg$. So if we could find a condition on a finitary 2-monad T on Cat for which every pseudo-T-structure on a category D is isomorphic to a strict-T-structure on D, that condition would suffice to ensure that T-structure transports along equivalence. So that is what we seek.

In fact, with subtle boot-strapping, such a condition may be seen as an instance of a phenomenon studied in [1], that being *flexibility*. So I shall explain the ideas surrounding flexibility, remark that it includes all our leading examples, and mention that in joint, as yet unpublished, work with Steve Lack, and acknowledging the ideas of Max Kelly, we can give a practical, general condition that includes our examples and that implies flexibility of a 2-monad.

First observe that, by a general result about enriched categories [2], the 2-category $End_f(Cat)$ of finitary endo-2-functors on Cat is locally finitely presentable, with a monoidal structure given by composition of 2-functors. A monoid in $End_f(Cat)$ is precisely a finitary 2-monad on Cat. Monoids in $End_f(Cat)$, together with (strict) maps of monoids and evident 2-cells, form a 2-category $Monoid(End_f(Cat))_s$, with $Monoid(End_f(Cat))_s = Monad_f(Cat)_s$.

For general reasons, this 2-category is finitarily monadic over $End_f(Cat)$, with 2-monad denoted by M. That allows us to boot-strap: $M\text{-}Alg_s = Monad_f(Cat)_s$; the 2-category $M\text{-}Alg$ is precisely the 2-category $Monad_f(Cat)$ of finitary 2-monads on Cat and pseudo-maps of 2-monads; and the inclusion J of $Monad_f(Cat)_s$ into $Monad_f(Cat)$ has a left adjoint we denote by $(-)'$.

So a finitary 2-monad T on Cat is precisely an M-algebra. In [1], an M-algebra T is called *flexible* if the counit $\epsilon_T : (JT)' \longrightarrow T$ has a section h in the 2-category $M\text{-}Alg_s$. It follows that, in the 2-category $M\text{-}Alg$, the map Jh is isomorphic to the unit $\eta_{JT} : JT \longrightarrow J(JT)'$. An M-algebra T is called *semi-flexible* if it satisfies the weaker condition, i.e., if there exists a map h in $M\text{-}Alg_s$ for which Jh is isomorphic to η_{JT}.

So how does this relate to our question? The short answer is that, if a 2-monad T is semi-flexible, every pseudo-T-algebra is isomorphic to a strict T-algebra on the same 0-cell. The argument goes as follows.

Assuming a little completeness, any object C of a category K, seen as a functor from 1 into K, extends by right Kan extension to an endofunctor we denote by $[C, C]$ on K. The functor $[C, C]$ has a canonical monad structure, and, given any monad T on K, to give a T-algebra structure on C is equivalent to giving a map of monads from T to $[C, C]$. One can routinely adapt this to deal with size and with pseudo-ness to allow K to be a 2-category, to make $[C, C]$ finitary, and so that to give a pseudo-T-algebra structure on C is equivalent to giving a pseudo map of monads from T to $[C, C]$.

A pseudo-T-structure on C is equivalent to a pseudo-map of 2-monads from T to $[C, C]$, thus, by adjointness, to a strict map from T' to $[C, C]$. By semi-flexibility, composition with h gives a strict map from T to $[C, C]$, thus a strict T-structure on C. As $h \cong \eta_{JT}$, this strict structure does the job.

References

[1] R. Blackwell, G.M. Kelly, and A.J. Power, Two-dimensional monad theory. J. Pure Appl. Algebra 59 (1989) 1–41.

[2] G.M. Kelly, Structures defined by finite limits in the enriched context 1, Cahiers de Top et Geom Differentielle 23 (1980) 3–42.

Input–Output Disjointness for Forward Expressions in the Logic of Information Flows

Heba Aamer
heba.mohamed@uhasselt.be
Universiteit Hasselt, Belgium

This is a joint work with Jan Van den Bussche which was published at ICDT 2021 [2].

Extended Abstract

FLIF (Forward Logic of Information Flows) [1] is an algebraic language for accessing atomic modules, and composing such modules into complex transition systems. Here, the term "atomic module" can be interpreted liberally: it can be a software library function, a Web service, a form on a website, or an information source.

Abstractly, an atomic module may be viewed as an n-ary relation where some of the arguments are designated as input arguments, with the remaining arguments providing output. For a simple example, a telephone directory may be viewed as a binary relation Dir(name; phone) with name as input argument and phone as output argument. For another example, the public bus company may provide its weekdays schedule as a relation Route(stop, interval; time, line, next, duration) that, given a bus stop and a time interval, outputs bus lines that stop there at a time within the interval, together with the duration to the next stop. Note how we use a semicolon to separate the input arguments from the output arguments.

In database research, such relations are known as information sources with limited access patterns, and the querying of such sources has received considerable attention. We refer to the research monograph by Benedikt et al. for more background [5]. An elegant syntactic fragment of first-order logic (FO) that takes into account the limitations imposed by the access patterns, was proposed by Nash and Ludäscher in the form of so-called "executable" FO formulas [8]. Executable FO queries can be evaluated by a form of relational algebra expressions, called *plans*, in which database relations can only be accessed by joining them on their input attributes with a relation that is either given as input or has already been computed.

FLIF now offers an alternative language, which we think of as situated halfway between executable FO and plans. In FLIF we take a novel graph-

based perspective to information sources with access patterns. The nodes of the graph are variable bindings; edges indicate accesses to the sources. For example, consider a source Friend(pname; fname) that outputs names of friends when given the name of a person as input. Then there is an edge labeled Friend from binding ν_1 to binding ν_2 if ν_2(fname) is a friend of ν_1(pname). Moreover, ν_2 should not differ from ν_1 in other variables (a principle of "inertia"). FLIF then is a simple XPath-like navigational query language over such graphs [9, 7, 4, 6, 10, 3].

For example, abbreviating Friend by F, the FLIF expression $F(x;y);F(y;z);F(z;u);(u=x)$ retrieves (in variable z) friends of friends of x who have x also as a direct friend. Here, the operator ; denotes composition and the subexpression $(u=x)$ serves as a test. Composition is a crucial operator by which (bounded-length) paths can be traced in the graph. FLIF also has union (to branch off), intersection (to merge branches) and difference (to exclude branches), and variable assignments.

Simple as FLIF may seem, it is at least as expressive as executable FO, as we showed together with Bogaerts, Surinx and Ternovska at ICDT 2020 [1]. Actually, executable FO was shown to be already subsumed by a well-behaved fragment of FLIF, formed by the expressions that are *io-disjoint*. IO-disjointness is a syntactic restriction on the *structure* of the expressions which guarantees that, during the transitions involved in evaluating the expression, the values of input variables are never overwritten. IO-disjoint expressions can also be evaluated by a very transparent translation to relational algebra plans, in which all joins are natural joins, and attribute renaming is not needed. (The example expression above is io-disjoint.)

Conversely, in our previous work we also gave a translation from io-disjoint FLIF to executable FO. It has remained open, however, whether every FLIF expression can actually be put in an equivalent form that is io-disjoint. In the paper [2] we answered this question affirmatively. Of course, we needed to make precise for what kind of "equivalence" this can work, and it was part of our contribution to clarify this. Intuitively, we showed that it is always possible to designate a fresh set of output variables disjoint from the set of input variables, in such a way that intermediate variables (used in subexpressions) do not interfere with either of the two sets. Proving this rigorously turned out to be a quite intricate task. Our result showed that io-disjoint FLIF is equally *powerful* as the full FLIF language. As a corollary, we obtain that full FLIF is not more powerful than executable FO.

References

[1] H. Aamer, B. Bogaerts, D. Surinx, E. Ternovska, and J. Van den Bussche. Executable first-order queries in the logic of information flows. In C. Lutz and J.C. Jung, editors, *Proceedings 23rd International Conference on Database Theory*, volume 155 of *Leibniz International Proceedings in*

Informatics, pages 4:1–4:14. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2020.

[2] H. Aamer and J. Van den Bussche. Input-output disjointness for forward expressions in the logic of information flows. In Ke Yi and Zhewei Wei, editors, *24th International Conference on Database Theory, ICDT 2021, March 23-26, 2021, Nicosia, Cyprus*, volume 186 of *LIPIcs*, pages 8:1–8:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[3] R. Angles, M. Arenas, P. Barceló, A. Hogan, J. Reutter, and D. Vrgoč. Foundations of modern query languages for graph databases. *ACM Computing Surveys*, 50(5):68:1–68:40, 2017.

[4] R. Angles, P. Barceló, and G. Rios. A practical query language for graph DBs. In L. Bravo and M. Lenzerini, editors, *Proceedings 7th Alberto Mendelzon International Workshop on Foundations of Data Management*, volume 1087 of *CEUR Workshop Proceedings*, 2013.

[5] M. Benedikt, J. Leblay, B. ten Cate, and E. Tsamoura. *Generating Plans from Proofs: The Interpolation-based Approach to Query Reformulation.* Morgan & Claypool, 2016.

[6] G.H.L. Fletcher, M. Gyssens, D. Leinders, D. Surinx, J. Van den Bussche, D. Van Gucht, S. Vansummeren, and Y. Wu. Relative expressive power of navigational querying on graphs. *Information Sciences*, 298:390–406, 2015.

[7] L. Libkin, W. Martens, and D. Vrgoč. Quering graph databases with XPath. In W.-C. Tan et al., editors, *Proceedings 16th International Conference on Database Theory*, pages 129–140. ACM, 2013.

[8] A. Nash and B. Ludäscher. Processing first-order queries under limited access patterns. In *Proceedings 23th ACM Symposium on Principles of Database Systems*, pages 307–318, 2004.

[9] J. Pérez, M. Arenas, and C. Gutierrez. nSPARQL: A navigational language for RDF. *Journal of Web Semantics*, 8(4):255–270, 2010.

[10] D. Surinx, G.H.L. Fletcher, M. Gyssens, D. Leinders, J. Van den Bussche, D. Van Gucht, S. Vansummeren, and Y. Wu. Relative expressive power of navigational querying on graphs using transitive closure. *Logic Journal of the IGPL*, 23(5):759–788, 2015.

Asymptotic Distribution of Parameters in Trivalent Maps and Linear Lambda Terms *

Olivier Bodini Alexandros Singh Noam Zeilberger

May 19, 2021

Abstract

Structural properties of large random maps and λ-terms may be gleaned by studying the limit distributions of various parameters of interest. In our work we focus on restricted classes of maps and their counterparts in the λ-calculus, building on recent bijective connections between these two domains. In such cases, parameters in maps naturally correspond to parameters in λ-terms and vice versa. By an interplay between λ-terms and maps, we obtain various combinatorial specifications which allow us to access the distributions of pairs of related parameters such as: the number of bridges in rooted trivalent maps and of subterms in closed linear λ-terms, the number of vertices of degree 1 in $(1, 3)$-valent maps and of free variables in open linear λ-terms etc. To analyse asymptotically these distributions, we introduce appropriate tools: a moment-pumping schema for differential equations and a composition schema inspired by Bender's theorem.

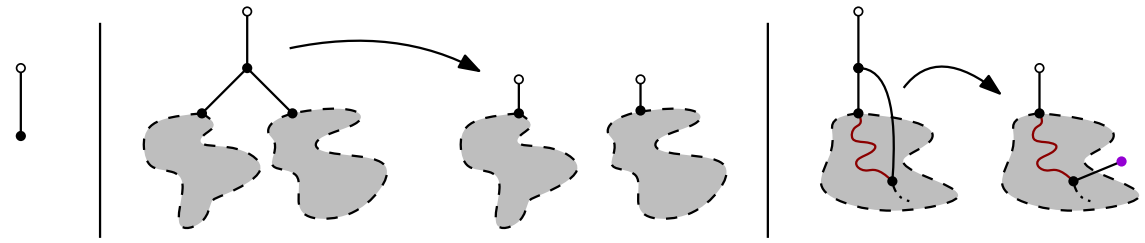
1 Introduction

Building upon an ever-increasing body of work on the combinatorics of maps, the λ-calculus, and their interactions, we present here a study of the asymptotic behaviour of some structural properties of large random objects drawn from restricted subclasses of maps and λ-terms.

1.1 Motivation and main results

Maps, or graphs embedded on surfaces, are an important object of study in modern combinatorics and their presence in various areas, ranging from algebra to physics, forms bridges between seemingly disparate subjects. In recent years it has become apparent that such bridges extend to logic as well, stemming from bijections between various natural classes of rooted maps and certain subsystems of λ-calculus. This includes a natural bijection between rooted trivalent maps and linear lambda terms [1], as well as a somewhat more involved bijection between rooted planar maps of arbitrary vertex degrees and β-normal ordered linear lambda terms [2], both of which have led to further study of the combinatorial interactions between lambda terms and maps.

To make the above correspondence concrete, let us briefly recall here the bijection of [1] following the analysis of [3], which is itself inspired by Tutte's classical approach to map enumeration via repeated root edge decomposition [4]. Informally, a rooted trivalent map may be defined as a graph equipped with an embedding into an oriented surface of arbitrary genus, all of whose vertices have degree 3, and one of whose edges has been distinguished and oriented (see below for a more formal definition). For reasons that will be quickly apparent, it is pertinent to slightly extend the class of rooted trivalent maps by embedding it into the class of (1,3)-valent maps, that is, maps whose vertices all have degree 3 or 1. We will view 1-valent vertices as labelled "external" vertices, and the root itself as a distinguished external vertex. By considering what happens around the root, it is clear that such a map falls into one of three categories:



*This is an extended abstract for a forthcoming paper. The coauthor presenting it would be Alexandros Singh.

namely, it is (from left to right) either the trivial one-edge map with no trivalent vertices and a single 1-valent vertex besides the root, a map in which the deletion of the root and its unique trivalent neighbour yields a pair of disconnected maps which may be canonically rooted, or finally a map in which the same operation yields a connected map which may be again rooted canonically and which in addition has a distinguished degree-1 vertex.

Quite remarkably, this decomposition à la Tutte exactly mirrors the standard inductive definition of linear lambda terms. Informally, an arbitrary lambda term is either a *variable* x, an *application* $(t\ u)$ of a term t to another term u, or an *abstraction* $\lambda x.t$ of a term t in a variable x, with linearity imposing the condition that in an abstraction $\lambda x.t$, the variable x has to occur exactly once in t. All of the terminology will eventually be explained, but concretely, the differential equation resulting from this analysis

$$T(z,u) = zu + zT(z,u)^2 + z\frac{\partial}{\partial u}T(z,u) \tag{1}$$

can be seen as counting either (1,3)-valent maps or linear λ-terms, with the size variable z tracking edges or subterms and the "catalytic" variable u tracking non-root 1-valent vertices or free variables. Setting $u = 0$ then allows us to recover the ordinary generating function $T(z,0)$ counting rooted trivalent maps in the classical sense as well as closed linear lambda terms.

The bijection from λ-terms to maps is made even more evident by representing the terms as certain decorated syntactic diagrams, in the manner of Figure 1. Such diagrams yield rooted trivalent maps with external 1-valent vertices simply by forgetting the labels of trivalent nodes, while the above correspondence shows that this information can be uniquely reconstructed from a given map by a recursive decomposition. A more comprehensive discussion of the correspondence between rooted trivalent maps and linear λ-terms is given in [1] and [3], and we will review it further below.

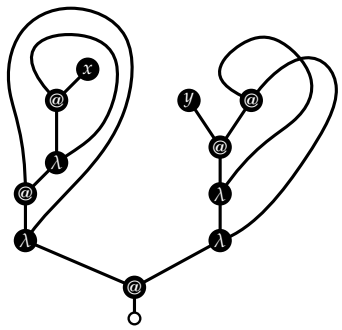


Figure 1: The syntactic diagram of the open linear λ-term $(\lambda a.(\lambda b.bx))(\lambda c.\lambda d.y(cd))$.

By exploiting such bijective correspondences between families of maps and λ-terms, we identify and study pairs of corresponding parameters natural to both classes, focusing on their limit distributions. The parameters studied in this work, together with their limit distributions, are listed in Table 1.

Parameter on maps (number of)	Parameter on λ-terms (number of)	Limit distribution
Loops in rooted trivalent maps	Identity-subterms in closed linear λ-terms	$Poisson(1)$
Bridges in rooted trivalent maps	Closed subterms in closed linear λ-terms	$Poisson(1)$
Vertices of degree 1 in rooted $(1,3)$-maps	Free variables in open linear λ-terms up to exchange	$\mathcal{N}\left((2n)^{1/3},(2n)^{1/3}\right)$
Vertices of degree 2 in rooted $(2,3)$-maps	Unused abstractions in closed affine λ-terms	$\mathcal{N}\left(\frac{(2n)^{2/3}}{2},\frac{(2n)^{2/3}}{2}\right)$

Table 1: Pairs of corresponding parameters in families of maps and λ-terms and their limit distributions, where $Poisson(\lambda)$ signifies the Poisson law of rate λ and $\mathcal{N}(\mu,\sigma^2)$ is shorthand for the corresponding random variables X_n converging in law to the standard normal distribution after being standardised as $\frac{X_n-\mu}{\sigma_n}$.

The first step of our approach is obtaining combinatorial specifications which allow us to capture the behaviour of our parameters of interest. This is done via a number of new decompositions valid for restricted families of maps and λ-terms. We are then faced with the task of asymptotically analysing these specifications, a task made difficult by the fact that number of elements of a given size in these families exhibits rapid growth; this precludes a straightforward approach based on standard tools of analytic combinatorics as the corresponding generating functions are purely formal power series and do not represent functions analytic at 0. To facilitate our approach, we therefore develop two new schemas which serve to encompass the two

general cases we have observed in our study: differential specifications giving rise to Poisson limit laws and composition-based specifications giving rise to Gaussian limit laws.

Our purpose in this present work is therefore twofold. On the one hand, we want to demonstrate how interesting insights on the typical structure of large random maps and λ-terms may be obtained by fruitfully making use of techniques drawn from the study of maps and λ-terms in tandem. On the other hand, we present two new tools which aid in the asymptotic analysis of parameters of fast-growing combinatorial classes; these tools are of independent interest, being applicable to the study of a wide class of combinatorial classes whose generating function is purely formal and obeys certain types of differential or functional equations.

1.2 Related work

The structure and enumeration, both exact and asymptotic, of maps by their genus has been the subject of much study; see, for example, [5, 6, 7] for the planar case and [8, 9, 10] for the higher genus case. On the other hand, the investigation of enumerative and statistical properties of rooted maps, counted without regard to their genus, has received much less attention. One reason for this is the divergent nature of the generating functions involved in such studies: by results derived in [11] one may show that the number of maps with n edges is asymptotically $(2n-1)!!$. This poses a significant obstacle since, as Odlyzko notes in [12]: "*There are few methods for dealing with asymptotics of formal power series, at least when compared to the wealth of techniques available for studying analytic generating functions*". As such, the structure of large random such maps has only recently begun to be investigated, starting with the distribution of genus in bipartite random maps being derived [13]. More recently, the authors of [14] investigated the asymptotic distributions for the number of vertices, root isthmic parts, root edges, root degree, leaves, and loops in random maps. In particular, comparing their results to ours, we note that for general maps the authors derived a $Poisson(1)$ limit law for the number of leaves and a previously-unknown law for the number of loops. Both of these results stand in stark contrast to the case of leaves in $(1,3)$-valent maps, which we show is normally distributed when standardised using $\mu = \sigma^2 = n^{1/3}$, and to the case of loops in rooted trivalent maps which we show is $Poisson(1)$. In terms of techniques employed, the authors of [14] show that for most of the statistics considered in their work the corresponding bivariate generating functions are formal solutions to Riccati equations, which may be linearised to yield recurrences on the coefficients of said generating functions which are amenable to study. We note here that an instance of a Riccati-type differential equation appears in our work too, but this time it is a differential equation with respect to the variable coupled to the statistic we're interested in, unlike the instances of [14] where the derivative was taken with regards to the size-coupled variable.

As for the λ-calculus, while of central importance to logic and theoretical computer science, it is a relatively new subject of study for combinatorialists. The combinatorial study of closed linear and affine λ-terms and their relaxations was introduced in [1, 15]. A comprehensive presentation of the combinatorics of open and closed linear λ-terms and their counterparts in maps is presented in [3]. We note here that there exists a number of combinatorial studies of λ-terms which use a size notion different than ours and that of [1, 15, 3]. For example, there exists a number of works focusing on a unary de Bruijn notation based model as in [16]. This choice of size notion has the effect of altering the qualitative properties of our objects of study: in particular, the statistical results and the associated techniques of [17] are not applicable to our model.

References

[1] O. Bodini, D. Gardy, and A. Jacquot, "Asymptotics and random sampling for BCI and BCK lambda terms," *Theoretical Computer Science*, vol. 502, pp. 227–238, 2013.

[2] N. Zeilberger and A. Giorgetti, "A correspondence between rooted planar maps and normal planar lambda terms," *Logical Methods in Computer Science*, vol. 11, no. 3:22, pp. 1–39, 2015.

[3] N. Zeilberger, "Linear lambda terms as invariants of rooted trivalent maps," *Journal of functional programming*, vol. 26, 2016.

[4] W. T. Tutte, "On the enumeration of planar maps," *Bulletin of the American Mathematical Society*, vol. 74, pp. 64–74, 1968.

[5] R. Cori and B. Vauquelin, "Planar maps are well labeled trees," *Canadian Journal of Mathematics*, vol. 33, no. 5, pp. 1023–1042, 1981.

[6] E. A. Bender and L. B. Richmond, "A survey of the asymptotic behaviour of maps," *Journal of Combinatorial Theory, Series B*, vol. 40, no. 3, pp. 297–329, 1986.

[7] C. Banderier, P. Flajolet, G. Schaeffer, and M. Soria, "Random maps, coalescing saddles, singularity analysis, and Airy phenomena," *Random Structures & Algorithms*, vol. 19, no. 3-4, pp. 194–246, 2001.

[8] T. R. Walsh and A. B. Lehman, "Counting rooted maps by genus. i," *Journal of Combinatorial Theory, Series B*, vol. 13, no. 3, pp. 192–218, 1972.

[9] D. Bessis, C. Itzykson, and J.-B. Zuber, "Quantum field theory techniques in graphical enumeration," *Advances in Applied Mathematics*, vol. 1, no. 2, pp. 109–157, 1980.

[10] M. Marcus and G. Schaeffer, "Une bijection simple pour les cartes orientables," Intern report A01-R-366 || marcus_01a, 2001. Rapport interne.

[11] D. Arques and J.-F. Béraud, "Rooted maps on orientable surfaces, riccati's equation and continued fractions," *Discrete mathematics*, vol. 215, no. 1-3, pp. 1–12, 2000.

[12] A. M. Odlyzko, "Asymptotic enumeration methods," *Handbook of combinatorics*, vol. 2, no. 1063, p. 1229, 1995.

[13] A. Carrance, "Uniform random colored complexes," *Random Structures & Algorithms*, vol. 55, no. 3, pp. 615–648, 2019.

[14] O. Bodini, J. Courtiel, S. Dovgal, and H.-K. Hwang, "Asymptotic distribution of parameters in random maps," in *29th International Conference on Probabilistic, Combinatorial and Asymptotic Methods for the Analysis of Algorithms (AofA 2018)*, vol. 110, pp. 13–1, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018.

[15] O. Bodini, D. Gardy, B. Gittenberger, and A. Jacquot, "Enumeration of generalized BCI lambda-terms," *Electron. J. Comb.*, vol. 20, no. 4, p. P30, 2013.

[16] M. Bendkowski, K. Grygiel, P. Lescanne, and M. Zaionc, "Combinatorics of λ-terms: a natural approach," *Journal of Logic and Computation*, vol. 27, no. 8, pp. 2611–2630, 2017.

[17] M. Bendkowski, O. Bodini, and S. Dovgal, "Statistical properties of lambda terms," *Electron. J. Comb.*, vol. 26, no. 4, p. P4.1, 2019.

Bisimulation between hom sets and logics without counting: Extended abstract

Nihil Shah

May 2021

Abstract

A classical result of Lovász [Lov67] states that two graphs G and H are isomorphic if and only if for every finite graph F, the number of homomorphisms from F to G is equal to the number of homomorphisms from F to H. Many results, of a similar form, hold by restricting the class of graphs to yield a weaker equivalence than isomorphism. In particular, in [Dvo09], counting homomorphisms from graphs of treewidth $\leq k$ yields equivalence under $k+1$-variable logic with counting quantifiers. Similarly, in [Gro20], counting homomorphisms from graphs of tree-depth $\leq n$ yields equivalence under first order logic with counting quantifiers and quantifier depth $\leq n$. Recent work [AKW21] has shown that these logics, without counting quantifiers, do not have a similar characterization. Inspired by the categorical formulation of Lovász-type results in [DJR21], and its use of the game comonad framework of Abramsky et. al [ADW17, AS21], we present ongoing work showing that a form of bisimulation between homomorphisms sets, rather than bijection, yield Lovász-type theorems for logics without counting.

Background

Formally, a classical result of Lovász [Lov67], which holds for general relational structures, states that for all σ-structures A, B in some relational vocabulary σ:

$$A \cong B \text{ if and only if } \forall \text{ finite } F, |\mathsf{Hom}(F, A)| = |\mathsf{Hom}(F, B)|$$

Weakenening the left-hand side of the above relation from isomorphism to equivalence in some resource-indexed logic results in the following Lovász-type theorems:

$$A \equiv^{C^{k+1}} B \text{ if and only if } \forall F \text{ w/ } \mathsf{treewidth}(F) \leq k, |\mathsf{Hom}(F, A)| = |\mathsf{Hom}(F, B)|$$
$$A \equiv^{C_n} B \text{ if and only if } \forall F \text{ w/ } \mathsf{tree\text{-}depth}(F) \leq k, |\mathsf{Hom}(F, A)| = |\mathsf{Hom}(F, B)|$$

where C^k and C_n are k-variable counting logic and counting logic with quantifier depth $\leq n$ (respectively). Both of these results are recovered in [DJR21] for "free", by proving (roughly) that for any comonad $\mathbb{C}$ over the category of σ-structures satisfying some conditions:

$$A \cong^{\mathcal{K}(\mathbb{C})} B \text{ if and only if } \forall \text{ finite coalgebras } F \to \mathbb{C}F, |\mathsf{Hom}(F, A)| = |\mathsf{Hom}(F, B)|$$

where $\mathcal{K}(\mathbb{C})$ is the Kleisli category of $\mathbb{C}$. To recover the Lovász-type theorems for logics with counting quantifiers, the above theorem is specialized to the Spoiler-Duplicator game comonads $\mathbb{P}_k$ and $\mathbb{E}_n$ introduced in [ADW17, AS18]. Namely, for the pebbling comonad $\mathbb{P}_k$, inspired by the k-pebble game, we have that isomorphism between structures A and B in $\mathcal{K}(\mathbb{P}_k)$ is equivalent to $A \equiv^{C^k} B$. Moreover, coalgebras $F \to \mathbb{P}_k F$ correspond to tree decompositions of width $< k$ for F, thereby demonstrating F has treewidth $< k$ [ADW17]. Similarly, for the Ehrenfeucht-Fraïssè comonad, isomorphism in $\mathcal{K}(\mathbb{E}_n)$ characterizes equivalence in first order

logic with counting quantifiers up to depth n, and coalgebras $F \to \mathbb{E}_n F$ demonstate that F has tree-depth $\leq n$ [AS18].

By contrast to the Lovász-type theorems for logics with counting quantifiers, in [AKW21], it was shown that there is no class of graphs $\mathcal{F}$ such that either of the following hold:

$$A \equiv^{L^{k+1}} B \text{ if and only if } \forall F \in \mathcal{F}, |\mathsf{Hom}(F, A)| = |\mathsf{Hom}(F, B)|$$
$$A \equiv^{L_n} B \text{ if and only if } \forall F \in \mathcal{F}, |\mathsf{Hom}(F, A)| = |\mathsf{Hom}(F, B)|$$

where L^k and L_n are k-variable logic and logic with quantifier depth $\leq n$ (respectively). Our work is motivated, in light of this result, to replace the relationship between sets of homomorphisms from bijection to some weaker notion. In particular, we believe that replacing bijection with some notion of open map bisimulation introduced in [JM94, JNW96] could yield a characterizations of $\equiv^{L^k}$ and $\equiv^{L_n}$ in terms of sets of homomorphisms.

Why Bisimulation?

We aim to prove such characterizations using the same methodology as used in [DJR21], i.e via a Spoiler-Duplicator game comonads.

In addition to characterizing equivalence for C^k and C_n, as well as coalgebraic definitions for treewidth and tree-depth, the Spoiler-Duplicator game comonads $\mathbb{P}_k$ and $\mathbb{E}_n$ also characterize equivalence for L^k and L_n. Namely, in [AS21], it was shown that:

$$A \equiv^{L^k} B \text{ if and only if } \exists \text{ span } \mathbb{P}_k A \leftarrow S \to \mathbb{P}_k B$$
$$A \equiv^{L_n} B \text{ if and only if } \exists \text{ span } \mathbb{E}_n A \leftarrow S \to \mathbb{E}_n B$$

where the legs of each span consists of open pathwise embeddings in the corresponding category of coalgebras. These spans are a modified notion of open map bisimulation as introduced in [JNW96]. This notion of bisimuation has been generalized from the cases of $\mathbb{P}_k$ and $\mathbb{E}_n$, and can be carried out in any so called arboreal category which has comonadic adjunction yielding a comonad $\mathbb{C}$ (called an arboreal cover) [AR21]. Therefore, following the example of [DJR21], we aim to prove for every arboreal cover $\mathbb{C}$:

$$\mathbb{C}A \leftarrow S \to \mathbb{C}B \text{ if and only if } \forall \text{ finite coalgbras } F \to \mathbb{C}F, \exists \text{ span } \mathsf{Hom}(F, A) \leftarrow \hat{S} \to \mathsf{Hom}(F, B) \quad (1)$$

Specializing to $\mathbb{P}_k$ and $\mathbb{E}_n$ would yield characterizations of $\equiv^{L^k}$ and $\equiv^{L_n}$ in terms of sets of homomorphisms.

Additional evidence for (1) comes from open maps between representable presheaves $\mathsf{Hom}(-, A)$ (in the sense of [JM94]) inducing open maps between transition systems (in the sense of [JNW96]). Having an open map between $\mathsf{Hom}(-, A)$ presheaves is much stronger than having an "open" map between $\mathsf{Hom}(X, A)$ sets. However, we believe just as Lovasz's classical result removes the naturality requirement from isomorphism between representable presheaves over graphs, so too (1), if established, would remove the naturality requirement for the correspondence between open maps of representable presheaves and open maps of transition systems (or in our case, general relational structures).

Conclusion

This talk will detail the ongoing work towards proving (1), both in the special cases of $\mathbb{E}_n$ and $\mathbb{P}_k$, and for any arboreal cover $\mathbb{C}$.

References

[ADW17] Samson Abramsky, Anuj Dawar, and Pengming Wang. The pebbling comonad in finite model theory. In *Proceedings of the 32nd Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '17. IEEE Press, 2017.

[AKW21] Albert Atserias, Phokion G. Kolaitis, and Wei-Lin Wu. On the expressive power of homomorphism counts, 2021.

[AR21] Samson Abramsky and Luca Reggio. Arboreal categories: An axiomatic theory of resources, 2021.

[AS18] Samson Abramsky and Nihil Shah. Relating structure and power: Comonadic semantics for computational resources. 2018.

[AS21] Samson Abramsky and Nihil Shah. Relating structure and power: Extended version, 2021.

[DJR21] Anuj Dawar, Tomáš Jakl, and Luca Reggio. Lovász-type theorems and game comonads, 2021.

[Dvo09] Zdeněk Dvořák. On recognizing graphs by numbers of homomorphisms. *Journal of Graph Theory*, 64(4):330–342, November 2009.

[Gro20] Martin Grohe. Counting bounded tree depth homomorphisms. In *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science*. ACM, July 2020.

[JM94] André Joyal and Ieke Moerdijk. A completeness theorem for open maps. *Annals of Pure and Applied Logic*, 70(1):51–86, November 1994.

[JNW96] André Joyal, Mogens Nielsen, and Glynn Winskel. Bisimulation from open maps. *Information and Computation*, 127(2):164–185, June 1996.

[Lov67] L. Lovász. Operations with structures. *Acta Mathematica Academiae Scientiarum Hungaricae*, 18(3-4):321–328, September 1967.

On the combinatorics of Girard's exponential

Peter M. Hines – University of York

Summary

We present a direct and straightforward link between some high-level abstract logical models, and some well-known combinatorics. Precisely, the representation of the exponential modality of J.-Y. Girard's linear logic [6] given in his first two Geometry of Interaction series of papers is shown to determine and be determined by the famous *Ruler Sequence* first seen in [7]; a well-known infinite, square-free, Ballot sequence that appears in fields as diverse as Gray Codes, the Towers of Hanoi problem, and cellular automata (see https://oeis.org/A001511 for the standard online resource, and [13] for a wide range of applications, and a good introduction generally). The correspondence between the two allows us to give an explicit form for the n-th term of this sequence, with applications such as describing Hamiltonian paths in hypercube graphs.

Background

The starting point for many introductions to linear logic is undoubtedly, based on [5], its *resource-sensitivity*. Propositions are thought of as resources that are consumed in the process of deduction, making it a sub-structural logic. However, the structural rules of weakening and contraction are not simply discarded; instead, they are controlled via a typing system that distinguishes between propositions that may copied, and those that may not. A modality !( ) known as the *exponential*, *bang*, or *of course* operation is used to promote propositions to a type that may be freely duplicated.

The conjunction and exponential in GoI

In the first two parts of Girard's Geometry of Interaction system [3, 4], propositions are represented by partial injections on the natural numbers[1] – i.e. elements of the symmetric inverse monoid $\mathcal{I}(\mathbb{N})$.

The (multiplicative) conjunction is represented by an injective inverse monoid homomorphism $\_\star\_ : \mathcal{I}(\mathbb{N}) \times \mathcal{I}(\mathbb{N}) \to \mathcal{I}(\mathbb{N})$, and the exponential is again an injective inverse monoid homomorphism $!(\_) : \mathcal{I}(\mathbb{N}) \to \mathcal{I}(\mathbb{N})$. These are given by, for all $f, g \in \mathcal{I}(\mathbb{N})$

- $f \star g = \begin{cases} 2f\left(\frac{n}{2}\right) & n \text{ even} \\ 2g\left(\frac{n-1}{2}\right)+1 & n \text{ odd.} \end{cases}$
- $!(f) = \Psi(Id_{\mathbb{N}} \times f)\Psi^{-1}$, where $\Psi : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ is the (monotone) Cantor-style bijection $\Psi(x, y) = 2^{x+1}y + 2^x - 1$

Direct calculation establishes that these are indeed injective inverse monoid homomorphisms, and for all $f \in \mathcal{I}(\mathbb{N})$ we have the required (right) fixed point property

$$f \star !(f) = !(f) \qquad \text{(The fixed-point condition)}$$

that allows us to treat $!(f) \in \mathcal{I}(\mathbb{N})$ as "an infinite number of copies of $f \in \mathcal{I}(\mathbb{N})$".

It is common (e.g. [1, 2, 8]) to interpret the above conjunction and exponential categorically, based on the observation that $\_\star\_$ is a semi-monoidal tensor in the sense of [12, 10], and $!(\_)$ is therefore a right fixed point functor for this tensor. However, we also wish to formalise the intuition that $!(f) = f \star (f \star (f \star (f \star \ldots)))$ is a well-defined limit, and use this to *derive* the bijection $\Psi : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$.

Conjunction and the bang via shuffling cards

A convenient intuition for the conjunction is in terms of *"shuffling infinite decks of cards"*. Glossing over the treatment of partiality, $f \star g$ may be thought of in the following terms :

[1] Readers more familiar with his system will recall that it was based on partial isometries on seperable Hilbert space. However, these arise from M. Barrs faithful functor $l_2 : \mathbf{pInj} \to \mathbf{Hilb}$ from the category of partial injections to the category of Hilbert spaces; as a consequence, the significantly simpler setting of partial injections suffices.

1. A single countably infinite deck of cards is dealt out in the usual way[2] to Alice and Bob.

2. Alice applies the permutation f to her stack of cards. Bob similarly applies permutation g to his stack.

3. Alice and Bob's hands of cards are then merged, using a perfect, interleaving, riffle shuffle.

In a similar way, the exponential may be thought of as being based around a shuffle of countably infinitely many decks of (countably infinitely many) cards. This arises as follows :

> "A pair of decks of cards is shuffled together, using a perfect riffle shuffle. However, the second of these two decks of cards arose from a perfect riffle shuffle of two decks of cards. The second of these two decks of cards also arose from a perfect riffle shuffle ..."

We formalise such descriptions by treating shuffles as (monotone) operations within the endomorphism operad of the natural numbers within the strict (semi-)monoidal category $(\mathbf{Bij}, \uplus)$ of bijections on Sets, with disjoint union.

In operadic terms, such a 'right-associated' shuffle of k decks of cards is given by a k-fold overproduct. The result of this acts by conjugation on the endomorphism operad of $\mathcal{I}(\mathbb{N})$ in a suitable category of inverse monoids, to give a "right-associated k-fold conjunction" $(\_ \star (\_ \star (\_ \star \ldots (\_ \star \_) \ldots))) : \mathcal{I}(\mathbb{N})^k \to \mathcal{I}(\mathbb{N})$.

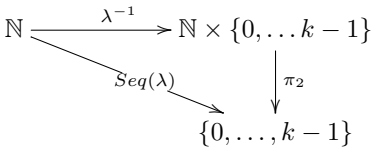
To give a meaning to infinitary (right-bracketed) conjunctions, including that required for the exponential, we simply need to demonstrate the convergence of infinitary overproducts within a suitable operad. This follows as a direct consequence of the monotonicity of the bijections used to model shuffles.

From bijections to sequences

The above considerations rely on the representation of shuffles as **bijections**. A shuffle of k decks of (countably infinite) cards is modeled as a monotone bijection $\lambda : \mathbb{N} \uplus \mathbb{N} \uplus \ldots \uplus \mathbb{N} \to \mathbb{N}$, or equivalently, $\lambda : \mathbb{N} \times \{0, \ldots, k-1\} \to \mathbb{N}$.

An alternative, entirely equivalent, way of describing shuffle is as **sequences**. The string $l_0 l_1 l_2 \ldots$ has the interpretation of *"the card placed at step number x comes from deck l_x"*. This is an operational description that tells us, practically, how to perform the shuffle in question.

The translation between these representations is straightforward. Given a monotone bijection $\lambda : \mathbb{N} \times \{0, \ldots, k-1\} \to \mathbb{N}$, the associated sequence $Seq(\lambda) : \mathbb{N} \to \{0, \ldots, k-1\}$ is given by the following composite



where $\pi_2 : \mathbb{N} \times \{0, \ldots k-1\} \to \{0, \ldots, k-1\}$ is simply the projection onto the second component. Such sequences are of course points of $\mathfrak{C}_{\{0,\ldots,k-1\}}$, the Cantor space over the set $\{0, \ldots, k-1\}$. The perfect interleaving riffle shuffle corresponding to Girard's conjunction determines / is determined by the *alternating Cantor point* $a(n) = n \ (mod\ 2)$ of the familiar binary Cantor space $\mathfrak{C}_{\{0,1\}}$.

It is natural to ask, which point of $\mathfrak{C}_{\mathbb{N}}$, the Cantor space over the natural numbers, corresponds to the shuffle used by the exponential?

The bang and the ruler sequence

Either from the description as an infinitary card shuffle, or by direct calculation, the sequence corresponding to Girard's bijection $\Psi : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ is the well-known **ruler sequence** (sequence number A001511 in the Online Encyclopedia of Integer Sequences) 0 1 0 2 0 1 0 3 0 1 0 2 0 1 0 4 0 1 0 2 0 1 0 3 0 1 0 2 0 1 0 ... characterised by the property that $r(n)$ is the number of divisors of $n+1$ of the form 2^k. Numerous other characterisations are given at https://oeis.org/A001511, including :

1. The number of trailing zeros in the binary representation of n

2. The number of the disk to be moved in the optimal solution of the Towers of Hanoi problem [9]

3. The Hamming distance between binary representations of n and $n+1$ [11].

[2] i.e. cards are alternately given to Alice and Bob. There is the minor difference that cards are taken from the bottom of the dealer's stack; $(\mathbb{N}, \leq)$ has a least element, but no top element.

Characterisation 1. gives us a concrete description of how to perform the shuffle of infinitely many decks of infinite cards : we simply count in binary, note at each step which column (labeled by 2^j, for some $j \in \mathbb{N}$) changes from 0 to 1, and play a card from the corresponding deck number j.

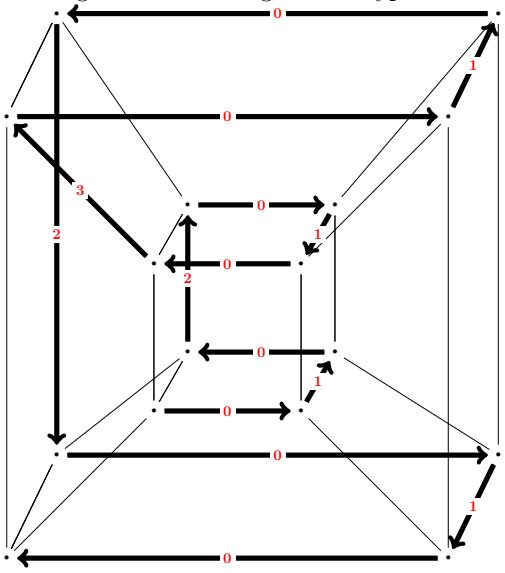
What is not given in OEIS is an explicit formula for $r(n)$. This may be recovered from the description of Girard's exponential in terms of card shuffling, giving $r = \pi_2 \Psi^{-1}$, where $\Psi(x, y) = 2^{x+1}y + 2^x - 1$. The fact that $r : \mathbb{N} \to \mathbb{N}$ is a *ballot sequence* (i.e. for all $x \in \mathbb{N}$, the number of occurrences of $x + 1$ in any prefix is less than or equal to the number of occurences of x) then follows directly from the fact that Ψ is monotone in both variables.

Another application of the Ruler sequence

We finish by giving a graphical interpretation of the ruler sequence that is not explicitly listed at OEIS, but is implicit in the well-known connection with Gray codes, and is also well-known in the field of Network Topologies : the ruler sequence (and hence, implicitly, Girard's bang) determines Hamiltonian circuits in hypercube graphs.

The first 2^n terms of the ruler sequence determine a Hamiltonian circuit for the 1-skeleton of the n-dimensional hypercube. Let us choose some labeling of our axes (i.e. dimensions) by $\{0, \ldots, k-1\}$, and pick some arbitrary vertex as the start. At step k, the next vertex in the path is found by moving along the unique edge that lies along axis $r(k)$. The path determined in this way then visits each vertex exactly once, and there is a (untraversed) edge linking the initial and final vertices, giving the desired Hamiltonian circuit. This is illustrated for the 4-dimensional hypercube in Figure 1, but the pattern is entirely general.

Figure 1: The bang on the hypercube



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

[4] 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.

[5] J.-Y. Girard, Y. Lafont, and P. Taylor. *Proofs and Types (2nd ed.)*. Cambridge University Press, 1990.

[6] Jean-Yves Girard. Linear logic: Its syntax and semantics. In *Proceedings of the Workshop on Advances in Linear Logic*, page 1–42, USA, 1995. Cambridge University Press.

[7] L. Gros. Théorie du baguenodier. *Aimé Vingtrinier*, 1872.

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

[9] Andreas M. Hinz, Sandi Klavzar, Uros Milutinovic, and Ciril Petr. *The Tower of Hanoi - Myths and Maths*. Birkhäuser, 2013.

[10] A. Joyal and J. Kock. Coherence for weak units. *Documenta Math.*, 18:71–110, 2013.

[11] Donald E. Knuth. *The Art of Computer Programming, Volume 1 (3rd Ed.): Fundamental Algorithms*. Addison Wesley Longman Publishing Co., Inc., USA, 1997.

[12] Joachim Kock. Elementary remarks on units in monoidal categories. *Mathematical Proceedings of the Cambridge Philosophical Society*, 144(1):53–76, 2008.

[13] Juan Carlos Nuño and Francisco J. Muñoz. On the ubiquity of the ruler sequence. *arXiv:2009.14629*, 2020.

Glued magic games self-test maximally entangled states

Thor Gabelgaard Nielsen

A major difference which sets quantum mechanics apart from classical theories is the existence of entangled states and with them the intrinsic randomness entailing that even though one knows exactly which state a quantum system is in, it is not always possible to know precisely which state subsystems are in. This is where all of quantum mechanic's power derives from, but the to harness it requires one to have a method of producing such states, and even if such a method is available, that comes with its own set of problems.

For example, if one is presented with an apparatus supposedly capable of producing entangled states, this cannot directly be verified as it is impossible to directly examine a state. Even if one constructs a measurement device and tries to verify by measurement, it suffers from the problem that it can only give a classical outcome, so the actual state cannot be determined with certainty. Furthermore, it introduces another possibility of error, namely that the measurement device could malfunction.

However, the fact that measuring some entangled states can give correlations stronger than classically attainable can obviously be used as a practical test for entanglement, since the required information is just the correlation in measurements, which is classical. Furthermore, if one observes a correlation stronger than attainable classically, it thus clearly stems from an entangled state, and therefore it is not necessary to place any kinds of assumptions upon the devices used for measurement. While this may not be immediately useful for testing for particular states, it turns out that certain correlations only arises from what is essentially a single state and measurements. If this is the case, that correlation is said to self-test the state and measurements, since by observing that correlation, one can be certain what the used state and measurements are, up to equivalence.

Therefore it is possible to consider a black-box model of experiments, where only the obtained correlations are used for concluding anything about the state of the system and the corresponding measurements used.

A convenient way of modelling such an experimental scenario is by the way of nonlocal games, which are played cooperatively between two players, which we denote by provers, and a referee. It proceeds by the referee drawing two questions from two finite sets of questions at random, sending one to each prover, and the provers, not allowed to communicate in any way has to send back their responses. The referee then checks whether they win the game according to some pre-defined condition.

For a given nonlocal game it is possible to analyse its maximal winning probability for some particular type of strategies. For many games, it is the case that some quantum strategies have a strictly higher winning probability than classical strategies, and such games can be used to certify the presence of entanglement. However, in some cases it is possible to say more, namely that all optimal quantum strategies are equivalent to a single ideal one, which we refer to as self-testing of that strategy. In our experimental scenario, the interpretation of this is clear; verifying that one is playing such a game optimally requires only classical information, and it certifies that a particular quantum state is present, along with the ideal measurements.

As an example of a nonlocal game, one can take the Mermin-Peres Magic Square game [Mer90; Per90], which is defined using a 3×3-board of integers. The goal is to fill this in such that each row and column sums to an even number, except for a single column which should have odd sum instead. This is cast as a nonlocal game by asking one prover to fill in a row or column and the other to fill in a single field of this row or column; they win if the row or column satisfies the appropriate constraint and they have assigned the same value to the field they both received.

As there does not exist an assignment satisfying the required constraints, there is no perfect classical strategy. Remarkably, there is a perfect quantum strategy [Ark12] using a state consisting of two EPR pairs. Furthermore, the game is a robust self-test of this strategy [WBMS16], in the sense that strategies winning with probability close to 1 also consists of a state and measurements close to the self-tested ones.

However, not all games are self-tests, as exemplified in [Cui+20], where the authors constructed a "Glued" Magic Square game by taking two Magic Square game boards, and stitching together the two odd constraints to form a single long column which should have odd sum, but otherwise leaving everything else unchanged. They exhibit two perfect strategies with inequivalent measurements, thus showing that their game is not a self-test. However, the two strategies still use equivalent states, and they pose the question of whether the game is a self-test for some state.

We positively answer that question and furthermore show that not only does the natural question of robustness also have a positive answer, but we also completely characterise all the optimal strategies. However, to capture the different possible strategies we introduce the concept of convex combinations of strategies, namely:

Definition. Suppose $\mathcal{S}_k = (\left|\psi^{(k)}\right\rangle, \{A_i^{(k)}\}_i, \{B_j^{(k)}\}_j)$ are all compatible quantum strategies, and $\{\alpha_k\}_{k=1}^n \subseteq [0,1]$ a set of scalars square-summing to 1. The *corresponding convex combination* is the strategy $\mathcal{S} = \sum_{k=1}^n \alpha_k \mathcal{S}_k$ formed by taking weighted direct sums of the states and measurement operators:

$$|\psi\rangle = \bigoplus_{k=1}^n \alpha_k \left|\psi^{(k)}\right\rangle, \qquad A_i = \bigoplus_{k=1}^n A_i^{(k)}, \qquad B_j = \bigoplus_{k=1}^n B_j^{(k)}.$$

Depending on one's viewpoint, this can be considered either as a decomposition of the state and operators, but it can also serve as a way to compose different strategies into a larger one. By taking the viewpoint of decomposing strategies, it is possible to naturally extend the definition of self-testing to convex combinations by replacing the concept of a single ideal strategy with a set of ideal strategies, where each optimal strategy decomposes to a number of substrategies, each equivalent to one of the ideal ones. By using this, we obtain our main theorem concerning convex self-testing of the Glued Magic Square.

Theorem (Informal)**.** *Suppose $\mathcal{S} = (|\psi\rangle \in \mathcal{H}_A \otimes \mathcal{H}_B, \{A_i\}_i, \{B_j\}_j)$ is a perfect strategy for the Glued Magic Square game. Then there exists a decomposition $\mathcal{S} = \alpha_1\mathcal{S}_1 + \alpha_2\mathcal{S}_2$ with both $\mathcal{S}_1$ and $\mathcal{S}_2$ containing subsets of operators forming a perfect strategy for the Magic Square game. Furthermore, the state of $\mathcal{S}_1$ and $\mathcal{S}_2$ are equivalent to the maximally entangled state of local dimension 4, $|\psi_4\rangle$, implying that there is a local isometry $U : \mathcal{H}_A \otimes \mathcal{H}_B \to \mathbb{C}^4 \otimes \mathbb{C}^4 \otimes \mathcal{H}_{A,aux} \otimes \mathcal{H}_{B,aux}$ and two quantum states $|aux_1\rangle, |aux_2\rangle$ such that*

$$U|\psi\rangle = (\alpha_1 |\psi_4\rangle \otimes |aux_1\rangle) \oplus (\alpha_2 |\psi_4\rangle \otimes |aux_2\rangle),$$

and from this one can factor out $|\psi_4\rangle$, which is thus self-tested.

Apart from this we also introduce a slightly more general notion of glued games and show that our methods also holds for these games.

Our main way of showing self-testing relies on reducing perfect strategies for the Glued Magic Square game to two perfect strategies for each of the constituent Magic Square games. The crucial point of this reduction is to note that if one has a perfect strategy for the Glued Magic Square game, then it assigns values to the stitched column such that they have odd sum. But splitting this column in two to get two Magic Square boards, one obtains an odd sum on one board and an even sum of the other. In this way, one gets a perfect strategy for the board with the odd sum. Thus, by subdividing the strategy into where it yields the odd sum, it is possible in the general case to reduce it to two perfect strategies for the Magic Square game.

This line of arguments can be extended to other games glued together in a similar way, and therefore we also obtain a similar result when one or both of the Magic Square parts are replaced with the highly related Magic Pentagram game, which is played on a pentagram-shaped board instead. However, a major obstacle to this reduction is to show that it is in fact well-defined, and that requires that the measurements used to assign values to the odd column is compatible with the other measurements used. We show that this is indeed the case both for the Magic Square and the Magic Pentagram, but these arguments do not easily generalise to larger classes of similar games.

Acknowledgements

This extended abstract is based upon a paper written in collaboration with Jitendra Prakash and Laura Mančinska.

References

[Ark12] Alex Arkhipov. 'Extending and characterizing quantum magic games'. In: *arXiv preprint arXiv:1209.3819* (2012).

[CM14] Richard Cleve and Rajat Mittal. 'Characterization of binary constraint system games'. In: *International Colloquium on Automata, Languages, and Programming*. Springer. 2014, pp. 320–331.

[CS17] Andrea Coladangelo and Jalex Stark. 'Robust self-testing for linear constraint system games'. In: *arXiv preprint arXiv:1709.09267* (2017).

[Cui+20] David Cui, Arthur Mehta, Hamoon Mousavi and Seyed Sajjad Nezhadi. 'A generalization of CHSH and the algebraic structure of optimal strategies'. In: *Quantum* 4 (2020), p. 346.

[KM17] Amir Kalev and Carl A Miller. 'Rigidity of the magic pentagram game'. In: *Quantum science and technology* 3.1 (2017), p. 015002.

[Mer90] Nathaniel David Mermin. 'Simple unified form for the major no-hidden-variables theorems'. In: *Physical review letters* 65.27 (1990), p. 3373.

[MPS21] Laura Mančinska, Jitendra Prakash and Christopher Schafhauser. 'Constant-sized robust self-tests for states and measurements of unbounded dimension'. In: *arXiv e-print arXiv:2103.01729* (2021).

[Per90] Asher Peres. 'Incompatible results of quantum measurements'. In: *Physics Letters A* 151.3-4 (1990), pp. 107–108.

[WBMS16] Xingyao Wu, Jean-Daniel Bancal, Matthew McKague and Valerio Scarani. 'Device-independent parallel self-testing of two singlets'. In: *Physical Review A* 93.6 (2016), p. 062121.

Big Ramsey degrees using Ramsey theorem for trees with interesting levels

Jan Hubička
Department of Applied Mathematics (KAM)
Charles University
Malostranské náměstí 25,
Praha 1, Czech Republic,
Email: hubicka@kam.mff.cuni.cz

***Abstract*—We discuss a new Ramsey theorem for trees with *interesting levels* which was developed in order to characterise the big Ramsey degrees of the homogeneous partial order. It can be seen as a joint strenghtening of the Milliken theorem for finitely branching trees and the Voigt lemma. This is a joint work with Balko, Chodounský, Dobrinen, Konečný, Vena and Zucker.**

I. Introduction

We use the standard notion of model-theoretic structures. Given structures $\mathbf{A}$ and $\mathbf{B}$, we denote by $\binom{\mathbf{B}}{\mathbf{A}}$ the set of all embeddings from $\mathbf{A}$ to $\mathbf{B}$. We write $\mathbf{C} \longrightarrow (\mathbf{B})^{\mathbf{A}}_{k,l}$ to denote the following statement: for every colouring χ of $\binom{\mathbf{C}}{\mathbf{A}}$ with k colours, there exists an embedding $f\colon \mathbf{B} \to \mathbf{C}$ such that χ does not take more than l values on $\binom{f(\mathbf{B})}{\mathbf{A}}$. For a countably infinite structure $\mathbf{B}$ and its finite substructure $\mathbf{A}$, the *big Ramsey degree* of $\mathbf{A}$ in $\mathbf{B}$ is the least number $l \in \mathbb{N}\cup\{\infty\}$ such that $\mathbf{B} \longrightarrow (\mathbf{B})^{\mathbf{A}}_{k,l}$ for every $k \in \mathbb{N}$.

A structure is *homogeneous* if every isomorphism between two of its finite substructures extends to an automorphism. The study of big Ramsey degrees of homogeneous structures started by result of Devlin who, in 1979 refining earlier result of Laver, characterised the big Ramsey degrees of linear orders in the order of rationals by showing that the big Ramsey degree of order of size n equals to the $(2n-1)$st derivative of the tangent function evaluated at 0 (see e.g. [1]). This interesting closed-form formula in fact counts special subtrees of the binary tree [2] (the Joyce trees).

Laflamme, Sauer and Vuksanović characterised big Ramsey degrees of the Rado (or random) graph and related random structures in binary languages [3]. Again the big Ramsey degrees corresponds to the number of special subtrees of the binary tree which were counted by Larson [2].

Trees used to characterise big Ramsey degrees follow naturally from the tree of 1-types of enumerations of the homogeneous structure in question. In both results above a key ingredient is the Milliken tree theorem [1] which is used to obtain the upper bound. Recently, new techniques for characterising big Ramsey degrees of structures with non-trivial forbidden substructures have been developed. In particular, a characterisation of big Ramsey degrees of the triangle-free Henson graph was obtained by Dobrinen [4] and independently by Balko, Chodounský, Hubička, Konečný, Vena and Zucker. It turns out that big Ramsey degrees provide an interesting new measure of the complexity of homogeneous structures (which are of interest in multiple areas including the Constraint Satisfaction Problems). See [5] for connection to topological dynamics.

II. Big Ramsey degrees of the generic partial order

It is well known that up to isomorphism there is a unique homogeneous partial order $\mathbf{P} = (P, \leq_{\mathbf{P}})$ such that every countable partial order has an embedding to $\mathbf{P}$. The order $\mathbf{P}$ is called the *generic partial order*.

Using parameter spaces the author [6] proved that the big Ramsey degree of any finite poset in $\mathbf{P}$ is finite. Here, we discuss a precise characterisation subsequently obtained by Balko, Chodounský, Dobrinen, Konečný, Vena and Zucker [7].

Our construction makes use of the following partial order introduced in [6]. Let $\Sigma = \{\mathrm{L}, \mathrm{X}, \mathrm{R}\}$ be an *alphabet* ordered by $<_{\mathrm{lex}}$ as $\mathrm{L}<_{\mathrm{lex}}\mathrm{X}<_{\mathrm{lex}}\mathrm{R}$. We denote by Σ^* the set of all finite words in the alphabet Σ, by $\leq_{\mathrm{lex}}$ their lexicographic order, and by $|w|$ the length of the word w (whose characters are indexed by natural numbers starting at 0). For $w, w' \in \Sigma^*$, we set $w \prec w'$ if and only if there exists i such that $0 \leq i < \min(|w|, |w'|)$, $(w_i, w'_i) = (\mathrm{L}, \mathrm{R})$, and $w_j \leq_{\mathrm{lex}} w'_j$ for every $0 \leq j < i$. It is not difficult to check that $(\Sigma^*, \preceq)$ is a partial order and that $(\Sigma^*, \leq_{\mathrm{lex}})$ is one of its linear extensions [6].

We write $w \perp w'$ if $w_i <_{\mathrm{lex}} w'_i$ and $w'_j <_{\mathrm{lex}} w_j$ for some $0 \leq i, j < \min(|w|, |w'|)$. (Note that it is not necessarily true that $(w \not\preceq w' \wedge w' \not\preceq w) \iff w \perp w'$. However, we will construct subsets of Σ^* where this is satisfied.) We call words w and w' *related* if one of expressions $w \preceq w'$, $w' \preceq w$ or $w \perp w'$ holds, otherwise they are *unrelated*.

Given a word w and an integer $i \geq 0$, we denote by $w|_i$ the *initial segment* of w of length i. For $S \subseteq \Sigma^*$, we let $\overline{S}$ be the set $\{w|_i \colon w \in S, 0 \leq i \leq |w|\}$. The set Σ^* can be seen as a rooted ternary tree and sets $S = \overline{S} \subseteq \Sigma^*$ as its subtrees. Given $i \geq 0$, we let $S_i = \{w \in S : |w| = i\}$ and call it the *level i of S*. A word $w \in S$ is called a *leaf* of S if there is no word $w' \in S$ extending w. Let $L(S)$ be the set of all leafs of S. Given a word w and a character $c \in \Sigma$, we denote by $w^\frown c$ the word created from w by adding c to the end of w. We also set $S^\frown c = \{w^\frown c : w \in S\}$.

Big Ramsey degrees in $\mathbf{P}$ are characterised by the following subsets of Σ^*:

Definition II.1 ([7])**.** A set $S \subseteq \Sigma^*$ is called a *poset-type* if $S = \overline{S}$ and precisely one of the following four conditions is satisfied for every i with $0 \leq i < \max_{w \in S} |w|$:

1) **Leaf:** There is $w \in S_i$ related to every $u \in S_i \setminus \{w\}$ and
$$S_{i+1} = (S_i \setminus \{w\})^\frown \mathrm{X}.$$
2) **Branching:** There is $w \in S_i$ such that
$$\begin{aligned} S_{i+1} &= \{z \in S_i : z <_{\mathrm{lex}} w\}^\frown \mathrm{X} \cup \{w^\frown \mathrm{X}, w^\frown \mathrm{R}\} \\ &\quad \cup \{z \in S_i : w <_{\mathrm{lex}} z\}^\frown \mathrm{R}. \end{aligned}$$
3) **New $\perp$:** There are unrelated words $v <_{\mathrm{lex}} w \in S_i$ s.t.
$$\begin{aligned} S_{i+1} &= \{z \in S_i : z <_{\mathrm{lex}} v\}^\frown \mathrm{X} \cup \{v^\frown \mathrm{R}\} \\ &\quad \cup \{z \in S_i : v <_{\mathrm{lex}} z <_{\mathrm{lex}} w \text{ and } z \perp v\}^\frown \mathrm{X} \\ &\quad \cup \{z \in S_i : v <_{\mathrm{lex}} z <_{\mathrm{lex}} w \text{ and } z \not\perp v\}^\frown \mathrm{R} \\ &\quad \cup \{w^\frown \mathrm{X}\} \cup \{z \in S_i : w <_{\mathrm{lex}} z\}^\frown \mathrm{R}. \end{aligned}$$
Moreover, the following assumption is satisfied:
(A) For every $u \in S_i$, $v <_{\mathrm{lex}} u <_{\mathrm{lex}} w$ implies that at least one of $u \perp v$ or $u \perp w$ holds.
4) **New $\preceq$:** There are unrelated words $v <_{\mathrm{lex}} w \in S_i$ s.t.
$$\begin{aligned} S_{i+1} &= \{z \in S_i : z <_{\mathrm{lex}} v \text{ and } z \perp v\}^\frown \mathrm{X} \\ &\quad \cup \{z \in S_i : z <_{\mathrm{lex}} v \text{ and } z \not\perp v\}^\frown \mathrm{L} \\ &\quad \cup \{v^\frown \mathrm{L}\} \cup \{z \in S_i : v <_{\mathrm{lex}} z <_{\mathrm{lex}} w\}^\frown \mathrm{X} \\ &\quad \cup \{w^\frown \mathrm{R}\} \\ &\quad \cup \{z \in S_i : w <_{\mathrm{lex}} z \text{ and } w \perp z\}^\frown \mathrm{X} \\ &\quad \cup \{z \in S_i : w <_{\mathrm{lex}} z \text{ and } w \not\perp z\}^\frown \mathrm{R}. \end{aligned}$$
Moreover, the following assumptions are satisfied:
(B1) For every $u \in S_i$ such that $u <_{\mathrm{lex}} v$, at least one of $u \preceq w$ or $u \perp v$ holds.
(B2) For every $u \in S_i$ such that $w <_{\mathrm{lex}} u$, at least one of $v \preceq u$ or $w \perp u$ holds.

Given a finite partial order $\mathbf{Q}$, we let $T(\mathbf{Q})$ be the set of all poset-types S such that $(L(S), \preceq)$ is isomorphic to $\mathbf{Q}$. As our main result, we determine the big Ramsey degrees of $\mathbf{P}$.

Theorem II.2 (Balko, Chodounský, Dobrinen, Konečný, Vena and Zucker [7])**.** *For every finite partial order $\mathbf{Q}$, the big Ramsey degree of $\mathbf{Q}$ in the generic partial order $\mathbf{P}$ equals $|T(\mathbf{Q})| \cdot |\mathrm{Aut}(\mathbf{Q})|$.*

One can view a poset-type S as a binary branching tree and each level S_i as a structure $\mathbf{S}_i = (S_i, \leq_{\mathrm{lex}}, \preceq, \perp)$. One can verify that if a level S_i is a leaf level, then $\mathbf{S}_{i+1}$ is isomorphic to $\mathbf{S}_i$ with one vertex removed. If a level S_i is a branching level, then $\mathbf{S}_{i+1}$ is isomorphic to $\mathbf{S}_i$ with one vertex $w \in S_i$ duplicated to $w^\frown \mathrm{X}, w^\frown \mathrm{R} \in S_{i+1}$. If a level S_i has new $\preceq$ (or $\perp$), then $\mathbf{S}_{i+1}$ is isomorphic to $\mathbf{S}_i$ extended by one pair in the relation $\preceq$ (or $\perp$).

III. Interesting levels and the lower bounds

Words $u \leq_{\mathrm{lex}} v$ are *compatible* if there is no $i < \min(|u|, |v|)$ such that $(u_i, v_i) = (\mathrm{R}, \mathrm{L})$, and if there exists $j < \min(|u|, |v|)$ such that $(u_j, v_j) = (\mathrm{L}, \mathrm{R})$ then $u \prec v$ and $u \not\perp v$. It can be checked that poset-types contains only mutually compatible words.

The following is a key definition for obtaining both lower and upper bounds on big Ramsey degrees. Given $S \subseteq \Sigma^*$, we call a level $\overline{S}_i$ *interesting* if the structure $\overline{\mathbf{S}}_i = (\overline{S}_i, \leq_{\mathrm{lex}}, \preceq, \perp)$ is not isomorphic to $\overline{\mathbf{S}}_{i+1} = (\overline{S}_{i+1}, \leq_{\mathrm{lex}}, \preceq, \perp)$ or there exist incompatible $u, v \in S_{i+1}$ such that $u|_i$ and $v|_i$ are compatible. Let $I(S)$ be the set of all interesting levels in $\overline{\mathbf{S}}$. Let $\tau_S \colon \overline{S} \to \Sigma^*$ be a mapping assigning every $w \in S$ the word created from w by deleting all characters with indices not in $I(S)$. Define $\tau(S) = \tau_S[S]$ and call it the *embedding type of S*.

Observation III.1. *For a poset-type S and $S' \subseteq L(S)$, $\tau(\overline{S'}) = \overline{\tau(S')}$ is a poset-type.* □

Theorem III.2 ([7])**.** *There exists an embedding $\psi \colon \mathbf{P} \to (\Sigma^*, \preceq)$ such that $S_{\mathbf{P}} = \overline{\psi[\omega]}$ is a poset-type and $\psi(i)$ is a leaf of $S_{\mathbf{P}}$ for every $i \in P$.*

Given a finite partial order $\mathbf{A}$, we construct, using Theorem III.2, a function (colouring) $\chi_{\mathbf{A}} \colon \binom{\mathbf{P}}{\mathbf{A}} \to T(\mathbf{A})$ by setting $\chi_{\mathbf{A}}(f) = \tau(\psi[f[A]])$ for every $f \in \binom{\mathbf{P}}{\mathbf{A}}$. It follows that $\chi_{\mathbf{A}}$ is an *unavoidable colouring* in the following sense:

Theorem III.3 ([7])**.** *For every finite partial order $\mathbf{A}$ and every $f \in \binom{\mathbf{P}}{\mathbf{P}}$, we have $\left\{\chi_{\mathbf{A}}[f \circ g] : g \in \binom{\mathbf{P}}{\mathbf{A}}\right\} = T(\mathbf{A})$.*

IV. The upper bounds and Ramsey theorem for trees with interesting levels

Historically, big Ramsey degrees have always been characterized after first obtaining very geneours upper bounds. After determining a lower bound believed to be tight, the proof for upper bounds was refined to give the precise characterisation. Quite surprisingly, the method in [6] does not seem to offer such a natural refinement and we had to develop a new Ramsey tree theorem we describe now.

Given $S \subseteq \Sigma^*$ we call function $f \colon S \to \Sigma^*$ *shape-preserving* if $\tau_S(w) = \tau_{f[S]}(f(w))$ and $|f(w)| \in I(f[S])$ for every $w \in S$.

We will generally consider shape-preserving functions only for those sets S such that $S = \tau(S)$ (that is for *embedding types*). The following follows directly from the definition:

Observation IV.1. *Let $f \colon S \to \Sigma^*$ be shape-preserving.*

(i) For every shape-preserving $h \colon f[S] \to \Sigma^*$ it holds that $h \circ f$ is shape-preserving.
(ii) For every $u, v \in S$, $|u| \leq |v|$ it holds that $|f(u)| \leq |f(v)|$.
(iii) For every $u, v \in S$ where u is an initial segment of v it holds that $f(u)$ is an initial segment of $f(v)$.
(iv) Function f is an embedding $f \colon (S, \leq_{\mathrm{lex}}, \preceq, \perp) \to (\Sigma^*, \leq_{\mathrm{lex}}, \preceq, \perp)$ and images of compatible words are also compatible.

Given $S \subseteq \Sigma^*$ and $\ell > 0$ we denote by $S|_{\leq \ell} = \cup_{i \leq \ell} S_i$ the set of all words in S of length at most ℓ. We also put $S|_{<\ell} = S|_{\leq \ell-1}$.

Remark. It is also possible to observe that for $S = \Sigma^*_\ell$ for some $\ell > 0$ it holds that shape-preserving functions corresponds to special strong subtrees as used by the Milliken's tree theorem.

Given $S, S' \subseteq \Sigma^*$ we denote by $\mathrm{Shape}(S, S')$ the set of all shape-preserving functions f such that $f[S] \subseteq S'$.

Let $S \subseteq \Sigma^*$. For shape-preserving function $g\colon S \to \Sigma^*$ we denote by $\widetilde{g}$ a function $\{|w| : w \in S\} \to \omega$ defined by $\widetilde{g}(i) = |g(w)|$ for some $w \in S$, $|w| = i$. (Note that by Observation IV.1 (ii) this is uniquely defined.)

Theorem IV.2. *For every finite $S = \overline{S} = \tau(S) \subseteq \Sigma^*$ and every finite colouring χ of $\mathrm{Shape}(\Sigma^*, S)$ there exists $f \in \mathrm{Shape}(\Sigma^*, \Sigma^*)$ such that $\chi \restriction \mathrm{Shape}(S, f[\Sigma^*])$ is constant.*

From Theorem IV.2 the upper bounds on big Ramsey degrees in $\mathbf{P}$ follow naturally:

Theorem IV.3. *For every finite partial order $\mathbf{Q}$ and every finite colouring χ of $\binom{\mathbf{P}}{\mathbf{Q}}$ there exists $f \in \binom{\mathbf{P}}{\mathbf{P}}$ such that $\left|\chi\left[\binom{f[\mathbf{P}]}{\mathbf{Q}}\right]\right| \leq |T(\mathbf{Q})|$.*

Proof. Fix a partial order $\mathbf{Q}$ and a colouring χ. Because $\mathbf{P}$ is generic there exists embedding $e\colon (\Sigma^*; \preceq) \to \mathbf{P}$. Define colouring χ' of $\binom{(\Sigma^*;\preceq)}{\mathbf{Q}}$ by putting $\chi'(g) = \chi(g \circ e)$. Now by a repeated application of Theorem IV.2 (see, for example [6]) it is possible to find a copy $f' \in \binom{(\Sigma^*;\preceq)}{(\Sigma^*;\preceq)}$ where a colour of $g' \in \binom{f'[(\Sigma^*;\preceq)]}{(\Sigma^*;\preceq)}$ depends only on $\tau(g' \circ e)$. Now $e^{-1}[f'[S_{\mathbf{P}}]]$ gives the desired copy of $\mathbf{P}$. □

V. Outline of proof of Theorem IV.2

Given $\ell > 1$ and $S \subseteq \Sigma^*_\ell$ we call function $e\colon \Sigma^*_\ell \to \Sigma^*_{\ell+1}$ an *extension* of S if for every $w \in \Sigma^*_\ell$ it holds that $e(w)$ extends w. Extension is *boring* if level ℓ of $e[S]$ is not interesting. We denote by Π_ℓ the set of all boring extensions of Σ^*_ℓ. If S is finite we denote by $\ell(S)$ the "last" level S_k where $k = \max_{w \in S} |w|$.

The key to proving the Ramsey property is the following correspondence between one level extensions and words in alphabet Π_ℓ.

Observation V.1. *Let $S = \overline{S} = \tau(S)$ be a finite subset of Σ^*, $\ell(S) = \{u^0 <_{\mathrm{lex}} u^1 <_{\mathrm{lex}} \cdots <_{\mathrm{lex}} u^n\}$ and $k = \max_{w \in S} |w|$. There is one-to-one correspondence between $\mathrm{Shape}(S, \Sigma^*)$ and pairs (g^0, w) where $g^0 \in \mathrm{Shape}(S_{<k}, \Sigma^*)$ and $w \in \Pi^*_{\ell(S)}$:*

1) *For every $g \in \mathrm{Shape}(S, \Sigma^*)$ there exists $w \in \Pi^*_{\ell(S)}$ such that for every $0 \leq i < n$ it holds that $g(u^i) = g(u^i|_{k-1}) \frown u^i_{k-1} \frown w^0_i \frown \cdots \frown w^{|w|-1}_i$.*
2) *Conversely, for every $g^0 \in \mathrm{Shape}(S_{<k}, \Sigma^*)$ and every word $w \in \Pi^*_{\ell(S)}$, the function $g'\colon S \to \Sigma^*$ defined by $g'(w) = g^0(w)$ for $|w| < k$ and $g'(u^i) = g^0(u^i|_{k-1}) \frown u^i_{k-1} \frown w^0_i \frown \cdots \frown w^{|w|-1}_i$ is shape-preserving.*

Given a finite alphabet Σ and $k \in \omega \cup \{\omega\}$, a *$k$-parameter word* is a (possibly infinite) string W in the alphabet $\Sigma \cup \{\lambda_i : 0 \leq i < k\}$ containing all symbols $\lambda_i : 0 \leq i < k$ such that, for every $1 \leq j < k$, the first occurrence of λ_j appears after the first occurrence of λ_{j-1}. The symbols λ_i are called *parameters*. Let W be an n-parameter word and let U be a parameter word of length $k \leq n$, where $k, n \in \omega \cup \{\omega\}$. Then $W(U)$ is the parameter word created by *substituting* U to W. More precisely, $W(U)$ is created from W by replacing each occurrence of λ_i, $0 \leq i < k$, by U_i and truncating it just before the first occurrence of λ_k in W. We apply the following Ramsey theorem for words, known as Voigh lemma, which an consequence of the Carlson–Simpson theorem [8], see also [6]:

Theorem V.2. *Let Σ be a finite alphabet. If Σ^* is coloured with finitely many colours, then there exists an infinite-parameter word W such that $W[\Sigma^*] = \{W(s)\colon s \in \Sigma^*\}$ is monochromatic.*

From Theorem V.2 the following lemma follows.

Lemma V.3 (Pigeonhole)**.** *Let $S = \overline{S} = \tau(S)$ be a finite subset of Σ^* containing at least one non-empty word. Put $k = \max_{w \in S} |w|$. Let $g^0 \in \mathrm{Shape}(S|_{<k}, \Sigma^*)$. Denote by G the set of all $g \in \mathrm{Shape}(S, \Sigma^*)$ extending g^0 and put $K = \widetilde{g^0}(k-1)$. Then for every finite colouring χ of G there exists $f \in \mathrm{Shape}(\Sigma^*, \Sigma^*)$ such that $f \restriction \Sigma^*_{\leq K}$ is identity and $\chi \restriction \mathrm{Shape}(S, f[\Sigma^*])$ is constant.*

Using the pigeonhole lemma the proof of Theorem IV.2 follows by the fusion argument. See i.e. [9].

Acknowledgment

The author is grateful to Matěj Konečný for remarks that improved quality of this abstract. The author is supported by the project 21-10775S of the Czech Science Foundation (GAČR). This article is part of a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 810115).

References

[1] S. Todorcevic, *Introduction to Ramsey spaces*. Princeton University Press, 2010, vol. 174.

[2] J. A. Larson, "Counting canonical partitions in the random graph," *Combinatorica*, vol. 28, no. 6, pp. 659–678, 2008.

[3] C. Laflamme, N. W. Sauer, and V. Vuksanovic, "Canonical partitions of universal structures," *Combinatorica*, vol. 26, no. 2, pp. 183–205, 2006.

[4] N. Dobrinen, "The Ramsey theory of the universal homogeneous triangle-free graph part II: Exact big Ramsey degrees," *arXiv preprint arXiv:2009.01985*, 2020.

[5] A. Zucker, "Big Ramsey degrees and topological dynamics," *Groups, Geometry, and Dynamics*, vol. 13, no. 1, pp. 235–276, 2019.

[6] J. Hubička, "Big Ramsey degrees using parameter spaces," *arXiv:2009.00967*, 2020.

[7] M. Balko, D. Chodounský, N. Dobrinen, J. Hubička, M. Konečný, L. Vena, and A. Zucker, "Big ramsey degrees of the generic partial order," 2021, extended abstract to EUROCOMB 2021, to appear in CRM Research Perspectives.

[8] T. J. Carlson and S. G. Simpson, "A dual form of Ramsey's theorem," *Advances in Mathematics*, vol. 53, no. 3, pp. 265–290, 1984.

[9] B. Voigt, "The partition problem for finite abelian groups," *Journal of Combinatorial Theory, Series A*, vol. 28, no. 3, pp. 257–271, 1980.

COMPLEXITY OF UNTYPED NORMALIZATION FOR SUBSYSTEMS OF LINEAR LAMBDA CALCULUS

ANUPAM DAS, DAMIANO MAZZA, LÊ THÀNH DŨNG NGUYỄN, AND NOAM ZEILBERGER

Abstract. We describe ongoing work on characterizing the complexity of the untyped β-equality and β-normalization problems for subsystems of linear lambda calculus defined by different syntactic restrictions with natural algebraic and graph-theoretic equivalents, including the non-symmetric/planar and non-unital/bridgeless fragments as well as their intersection.

1. Introduction

Pure lambda calculus is an exceedingly elegant computational artifact, with all of its power seemingly derived from a single rule, β-reduction $(\lambda x.t)(u) \to t[u/x]$. Determining whether a general lambda term has β-normal form is undecidable, and in fact historically this was the first published example of an undecidable problem [Chu36]. If one restricts to simply-typed terms then strong normalization holds, but deciding whether two terms are equal modulo β-conversion still has non-elementary complexity [Sta79].

We are interested in the untyped normalization problem for different subsystems of linear lambda calculus, that is, subsystems of pure lambda calculus defined by (at least) the restriction that every variable is used exactly once. It is known that deciding β-equality of linear terms is complete for polynomial time, a result due to Mairson [Mai04]. Indeed, since the size of a term decreases with every β-reduction (one application, one abstraction, and one variable are removed) it is easy to see that normalization can be performed in polynomial time, but conversely, Mairson established **P**-hardness by reduction from the circuit value problem, building up linear terms to simulate the evaluation of boolean circuits. (See also Terui's closely related and near simultaneous proof of **P**-completeness for normalization of MLL proof nets [Ter04].) We focus on the lattice of subsystems defined by selectively imposing one or more of the following additional restrictions on linear lambda terms:

(O) variables must be used in order, in other words the exchange rule is disallowed (for example, the term $\lambda x.\lambda y.\lambda z.x(yz)$ satisfies O but $\lambda x.\lambda y.\lambda z.(xz)y$ does not);
(V) every subterm must have a free variable, in other words closed subterms are disallowed (for example, the term $\lambda y.\lambda z.(xz)y$ satisfies V but $\lambda y.x(\lambda z.z)y$ does not).

These restrictions have both algebraic and topological motivations. Algebraically, as is well known, linear lambda terms may be interpreted as morphisms in any symmetric monoidal closed category [BS11], or even in a (non-monoidal) symmetric closed category. Linear terms satisfying condition O may then be interpreted in the wider class of *non-symmetric* closed categories, while terms satisfying V may be interpreted in *non-unital* symmetric closed categories. Topologically, it is folklore that linear lambda terms may be faithfully represented by a certain kind of proof-net [Mai02], and more recently it has been observed that this correspondence can be strengthened to a size-preserving bijection between linear lambda terms and rooted 3-valent graphs embedded on oriented surfaces of arbitrary genus (or "rooted 3-valent maps" for short [BGJ13, Zei16]). Then, restricting the bijection to O-terms yields exactly the *planar* rooted 3-valent maps, while restricting to V-terms yields exactly the *bridgeless* rooted 3-valent maps.

Conditions O and V define proper subsystems of linear lambda calculus in the sense that both properties are preserved under β-reduction. However, as far as we know, nothing is known about the complexity of deciding β-equality or normalization for these fragments, nor for their intersection, beyond the trivial polynomial time upper bound sketched above that applies for general linear terms.

Date: June 19, 2021.
SMP2021 note: this is a proposal for a talk about unpublished work-in-progress. Noam will be presenting.

The O fragment was briefly discussed by Abramsky [Abr08] under the heading of "planar lambda calculus", and to avoid confusion we will adopt the topological terminology (which is hopefully pretty evocative) in the sequel, referring to O-terms as planar terms and V-terms as bridgeless terms. Extensions of planar lambda calculus have also been studied within the broader setting of non-commutative/ordered linear logic (cf. [Pol01]), and it is worth mentioning that such a calculus was used recently by Nguyễn and Pradic to obtain a characterization of star-free languages within the framework of implicit complexity, relying on the O constraint in an essential way [NP20]. Bridgeless lambda calculus has been studied less, although it is also worth mentioning that Lambek's original syntactic calculus [Lam58] included versions of both restrictions O and V, in the sense that not only did his sequent calculus not have an exchange rule, but also it was restricted to sequents with a non-empty left-hand side.

2. Formal problem statements

We consider the complexity of two kinds of problems related to untyped normalization:

(1) Given a term t, what is the β-normal form of t? (β-normalization)
(2) Given two terms t_1 and t_2, is $t_1 =_\beta t_2$? (β-equality)

The second problem of course may be reduced to the first by computing the normal forms of t_1 and t_2 and checking that they are equal, but in the converse direction there isn't an obvious reduction. As formulated, (1) is a function problem rather than a decision problem – however, it can be turned into a decision problem while retaining the property that (2) can be reduced to it by reformulating (1) as a question of verifying whether a given "term with holes" is an approximation of the β-normal form of t. Thus formulated, we can view both β-normalization and β-equality as decision problems and try to characterize their complexity using standard decision-theoretic complexity classes like **P** and **L**.

In order to make formal statements about complexity, we need to add some precision about the representation of lambda terms as strings. The exact details are not so important, but we can distinguish two broad categories of representation: ones based on a serialization of the term tree (e.g., standard λ-calculus notation, or postfix notation), which we refer to as "tree-based" representations, and ones based on a labelled encoding of the term graph (e.g., proof-nets), which we refer to as "graph-based". The distinction is important when talking about the complexity class **L** or below, since the conversion from graph-based to tree-based representations typically requires at least logarithmic space.

The difference between typed and untyped normalization is subtle in the linear setting, since every linear term is simply typable, and its β-normal form is uniquely determined by its principal type [Hir93]. By *typed normalization* is meant that a typing derivation is provided in some suitable format as part of the input and output, whereas in *untyped normalization* no such typing derivation is provided, only the term itself. We emphasize that we are considering untyped normalization, although we are also interested in typability.

3. Results

As already mentioned, Mairson proved that deciding β-equality of linear lambda terms is **P**-hard and hence **P**-complete, by reduction from the circuit value problem (CVP). His encoding relied on a violation of the O-condition in an essential way, with the booleans True and False encoded as $\lambda x.\lambda y.\lambda k.kxy$ and $\lambda x.\lambda y.\lambda k.kyx$ respectively, and we were thus quite surprised to find out that:

Result 1. *The untyped β-equality problem for planar lambda terms is* **P***-complete.*

Proof sketch. The high-level structure of the proof is the same as Mairson's, giving a reduction of CVP to deciding β-equality, but with some key variations. First and most important is the encoding of booleans, and in fact we found two different possible encodings, both a simple-minded one where False and True are $\lambda x.x$ and $\lambda x.x(\lambda y.y)$, respectively, and a more conceptual one where they are $\lambda k.\lambda f.kf(\lambda x.x)$ and $\lambda k.\lambda f.k(\lambda x.x)f$, respectively. A benefit of the latter encoding is that the booleans may be assigned a simple type, and it turns out they are closely related to the type of booleans introduced by Matsuoka in an alternative proof of Mairson's theorem [Mat15]. With either encoding of the booleans, we can find closed planar terms $And, Not, \ldots$, representing the basic logic gates satisfying equations like

$$And\ True\ True =_\beta True \qquad And\ True\ False =_\beta False \qquad \ldots$$

as well as a closed planar term *Copy* representing a "fan-out" gate and which satisfies the equation $Copy\ B =_\beta \lambda f.f\ B\ B$ for $B \in \{True, False\}$ and then go on to build up circuits in a manner similar to Mairson. Still, there is another twist that with this set of gates, the O-condition means that we can only build *planar* circuits. However, it is well known that CVP may be reduced to planar CVP, and indeed we can explicitly construct a *Swap* gate as a planar term satisfying the equation $Swap\ B_1\ B_2 =_\beta \lambda f.f\ B_2\ B_1$ for $B_1, B_2 \in \{True, False\}$. With this additional gate in our toolkit we can then directly encode any boolean circuit C as a planar term t_C and reduce CVP to deciding $t_C =_\beta True$. □

Both Mairson's original linear encoding and our planar encoding of CVP include bad violations of the V-condition, and at first sight it appears difficult to do any serious computations with bridgeless terms. Nevertheless we have the following:

Result 2. *The untyped β-normalization problem for bridgeless lambda terms is* **P***-complete.*

The proof relies on a translation of general linear terms t to bridgeless terms t^V, which wraps variables of the original term with "handlers" that intercept any eventual applications. If we normalize t^V then we obtain a term with remnants of handlers interspersed throughout, but these can be easily removed in a postprocessing phase to obtain the β-normal form of t. Since β-normalization of linear terms is **P**-hard and all of these transformations can be done in polynomial time, this establishes the claim. Something that leaves us uneasy, though, is that we do not see how to adapt this approach to prove the stronger claim that β-equality of bridgeless terms is **P**-complete.

Finally, we are left with the intersection of the two conditions O and V, and here the situation is even more unresolved. It seems very difficult to program with bridgeless planar terms, and for now we have only been able to establish the following hardness reductions (note the dependence on the representation):

Result 3. *The untyped β-equality problem for bridgeless planar lambda terms is* **L***-hard on the graph representation, and* $\mathbf{TC}_0$*-hard on the tree representation.*

On the other hand, despite our attempts, we have not been able to conceive of an algorithm to decide β-equality in any class below **P**, using either representation.

References

[Abr08] Samson Abramsky. Temperly-Lieb algebra: from knot theory to logic and computation. In *Mathematics of Quantum Computing and Technology*, Applied Mathematics and Nonlinear Science Series, pages 415–458. Chapman and Hall/CRC, 2008.

[BGJ13] O. Bodini, D. Gardy, and A. Jacquot. Asymptotics and random sampling for BCI and BCK lambda terms. *Theoretical Computer Science*, 502:227–238, 2013.

[BS11] John Baez and Mike Stay. Physics, topology, logic and computation: A Rosetta stone. In B. Coecke, editor, *New Structures for Physics*, volume 813 of *Lecture Notes in Physics*, pages 95–174. Springer, 2011.

[Chu36] Alonzo Church. An unsolvable problem of elementary number theory. *Am. J. Math.*, 58(2):345–363, 1936.

[Hir93] Sachio Hirokawa. Principal types of bck-lambda-terms. *Theoretical Computer Science*, 107(2):253–276, 1993.

[Lam58] Joachim Lambek. The mathematics of sentence structure. *The American Mathematical Monthly*, 65(3):154–170, 1958.

[Mai02] Harry G. Mairson. From Hilbert spaces to Dilbert spaces: Context semantics made simple. In Manindra Agrawal and Anil Seth, editors, *FST TCS 2002: Foundations of Software Technology and Theoretical Computer Science, 22nd Conference Kanpur, India, December 12-14, 2002, Proceedings*, volume 2556 of *LNCS*, pages 2–17. Springer, 2002.

[Mai04] Harry G. Mairson. Linear lambda calculus and PTIME-completeness. *J. Functional Programming*, 14(6):623–633, November 2004.

[Mat15] Satoshi Matsuoka. A new proof of P-time completeness of linear lambda calculus. In Ansgar Fehnker, Annabelle McIver, Geoff Sutcliffe, and Andrei Voronkov, editors, *20th International Conferences on Logic for Programming, Artificial Intelligence and Reasoning - Short Presentations, LPAR 2015, Suva, Fiji, November 24-28, 2015*, volume 35 of *EPiC Series in Computing*, pages 119–130. EasyChair, 2015.

[NP20] Lê Thành Dung Nguyên and Pierre Pradic. Implicit automata in typed λ-calculi I: aperiodicity in a non-commutative logic. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 168 of *LIPIcs*, pages 135:1–135:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[Pol01] Jeff Polakow. *Ordered linear logic and applications*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, 2001.

[Sta79] Richard Statman. The typed λ-calculus is not elementary recursive. *Theoretical Computer Science*, 9:73–81, 1979.

[Ter04] Kazushige Terui. Proof nets and boolean circuits. In *19th IEEE Symposium on Logic in Computer Science (LICS 2004), 14-17 July 2004, Turku, Finland, Proceedings*, pages 182–191. IEEE Computer Society, 2004.

[Zei16] Noam Zeilberger. Linear lambda terms as invariants of rooted trivalent maps. *J. Functional Programming*, 26, 2016.

Traversal-invariant definability and Logarithmic-space computation

Siddharth Bhaskar — *University of Copenhagen* — *sbhaskar@di.ku.dk*
Steven Lindell — *Haverford College* — *slindell@haverford.edu*
Scott Weinstein — *University of Pennsylvania* — *weinstein@cis.upenn.edu*

Abstract

In the presence of arithmetic, order-invariant definability in first-order logic captures constant parallel time, uniform AC^0 [I]. The ordering is necessary to replicate the presentation of a structure on the input tape of a machine. What if that ordering was in fact a traversal of the structure? We show that an analogous notion of traversal-invariant definability characterizes deterministic logarithmic space (L) and that breadth-first traversals characterize nondeterministic logarithmic space (NL). The surprising feature of these characterizations is that we can work entirely within the framework of first-order logic without any extensions, other than the traversals themselves.

Keywords: first-order logic; logarithmic space computation; order invariant definability; graph traversals

Introduction

In the descriptive approach to complexity, formulas in extensions of first-order logic used to express the solution to a problem replace algorithms running on resource-bounded models of machine computation. Those extensions have relied on variations of classically developed inductive definability, known as fixed-point logic [M]. For example, a query on finite ordered graphs is expressible in fixed-point logic if and only if it is computable in polynomial-time [I]. Similarly, various transitive-closure logics capture logarithmic-space computability on ordered structures [I].

Order Invariant definability

Central to the connection between logic and complexity is understanding the precise role of order, since whenever a structure is given to a machine, it must be presented in some order on the input tape. A machine that purports to compute an isomorphism invariant graph query must output an answer which is independent of that order. Order-invariant first-order logic attempts to capture this notion with an arbitrary order of the input. While this paper does not address the role of that order in general, it does address the specific role that special traversal orderings play in invariant first-order definability.

Definition: A sentence σ (more generally a formula) is *order-invariant* over finite ordered graphs if for every finite graph G and any two linear orderings $<_1$ and $<_2$ of it, $(G, <_1)$ satisfies σ iff $(G, <_2)$ satisfies σ.

Note: A sentence over ordered graphs is order-invariant in the finite just in case it is invariant under adjacent transpositions because every finite permutation can be written as the product of adjacent transpositions.

Yuri Gurevich has shown order invariant logic is a nontrivial extension of first-order logic, though connectivity still cannot be defined in this setting because it is still local [L]. But what if instead of assuming our structures are presented in an arbitrary order, we assume they are presented in a particularly natural order? One such notion is that of a traversal order, classically defined for connected graphs [D], easily extended to arbitrary simple graphs, and even arbitrary structures via their Gaifman graph.

Definition: A *traversal* of a connected graph is a linear ordering of its vertices in which every initial segment is connected. An arbitrary graph traversal is an arbitrary ordering of its individual component traversals. A traversal of a relational structure S is simply a traversal of its Gaifman graph, the simple graph which connects two distinct elements of S whenever they occur together in some tuple of a relation from S.

Breadth-first search yields standard examples of graph traversals.

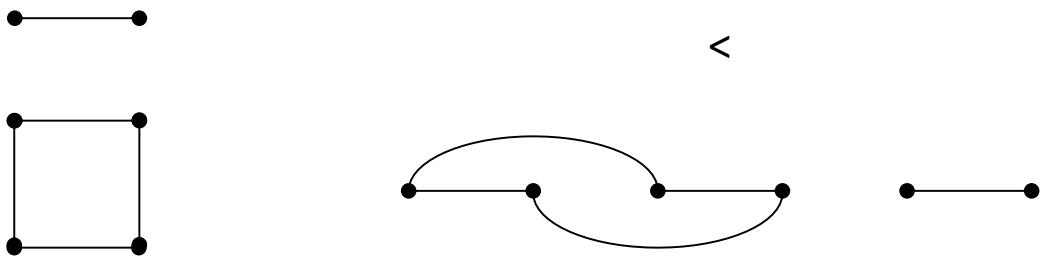


Figure: a graph (left) and one of its traversal presentations (right)

A traversal of a connected graph is characterized by the property that every node has a preceding neighbor. This fact can be used to show that the property of being a traversal is itself elementary [CK].

Traversal invariant definability

Suppose we take first-order formulas on ordered graphs which are invariant of the order, whenever that order is a valid traversal of the graph. In other words, the formula determines a query on graphs given any traversal of the graph.

Definition: A sentence σ (more generally a formula) is *traversal-invariant* over finite ordered graphs if for every finite graph G and any two traversals $<_1$ and $<_2$ of it, $(G, <_1)$ satisfies σ iff $(G, <_2)$ satisfies σ.

Examples: A traversed graph is connected iff every element except the first has a preceding neighbor. A traversed graph is acyclic if and only if it does not have a node with two preceding neighbors. Undirected reachability is traversal invariant because a smaller node x can reach a larger node y just in case every node in the interval $(x, y]$ has a preceding neighbor.

Closing the idea of traversal invariance under first-order interpretations leads us to define traversal invariant queries as compositions of a first-order translation π, a traversal $<$ of the resulting structure, and finally a traversal invariant sentence σ. Since the traversal of an arbitrary structure is just a traversal of its Gaifman graph, and since the map from an arbitrary structure to its Gaifman graph is first-order definable, we can without loss of generality assume that π produces that simple graph and assume that σ is a formula over ordered simple graphs. In pictures (the traversal is the key middle step):

$$S \to \pi(S) = G \to (G, <) \to \sigma(G, <)$$

Definition: A (Boolean) query is *traversal invariant* if it can be defined by $\sigma(\pi(S), <)$ where π is a first-order interpretation, and σ is a first-order sentence over ordered simple graphs with the property that for any traversal $<$ of $\pi(S)$, whether $(\pi(S), <)$ satisfies σ is independent of the traversal $<$ chosen. In other words, σ determines a well-defined query using any traversal $<$ of $\pi(S)$ satisfying τ. There is an obvious generalization to non-Boolean queries when σ is a formula which we omit here.

The naturalness of a traversal presentation cannot be overemphasized: it captures the notion that whenever possible, a new piece of input data is related to some previous piece. One of the most familiar examples of a traversal is breadth-first search. However, it is almost always described procedurally, whereby a method is explained to number and thereby order the vertices according to the BFS algorithm. But a logical presentation helps to reveal its fundamental properties.

Definition: A *breadth-first traversal* is one in which the earliest neighbor function (the parent in the breadth-first tree) is monotone (let the parent of the first element in a component be itself). It is easy to see that this captures the queue property of the breadth-first algorithm, since if the vertices are laid out in a line

corresponding to their breath-first order, none of the edges in the breadth-first tree will nest, one properly inside the other. Clearly, this property is elementary (first-order definable).

Results

As mentioned earlier, for order invariant first-order logic to capture uniform-AC^0, we need to assume that our logic also has access to arithmetic over that order, which is equivalent to FO(<, *bit*) [I]. Since the bit predicate is nothing more than the set membership relation over hereditarily finite sets in reverse, we define the notion of situated structures analogous to that of ordered structures.

Definition: Call a structure *S situated* if its signature includes an accompanying bit predicate.

These structures are without loss of generality already ordered, since < is elementarily definable from *bit* [DDLW]. We can now state our first result, which characterizes logarithmic-space computation (L).

Theorem: A query on situated structures is computable in L iff it is traversal invariant.

Our second result characterizes non-deterministic log-space computability (NL) in terms of breadth-first search. NL is the class of breadth-first traversal-invariant queries, defined like traversal invariance before except that we are guaranteed the given ordering of the graph is in fact a breadth-first traversal.

Theorem: A query on situated structures is computable in NL iff it is breadth-first invariant.

One might think that including the bit predicate in the initial input is important. But by using the equivalence between logarithmic-space machines and multi-head finite automaton, we can improve both above results in that we can capture queries on structures that come only with a successor relation and do not require the bit relation. The details of this are quite technically involved and omitted from this abstract.

Extension: A query on structures with successor is in (N)L iff it is (breadth-first) traversal invariant.

Conclusion

We have characterized logarithmic-space computability in terms of traversal invariant first-order definability in logic. Other ordered presentations might allow for analogous elementary characterizations of NC^1 or even P. In general, presentation invariant definability stays inside of NP ∩ co-NP [GLL]. We conclude with the following tantalizing question regarding traversal invariant definability.

Conjecture: Is polynomial-time computability characterized by depth-first invariant definability?

References

[CK] Corneil, D. & Krueger, R. "A Unified View of Graph Searching" SIAM J. Discrete Math., 22(4), 1259–1276.

[D] Diestel, R. *Graph Theory*, 4th Edition, Springer, 2012.

[DDLW] Dawar, A. & Doets, K. & Lindell, S. & Weinstein, S. "Elementary Properties of the Finite Ranks" Mathematical Logic Quarterly, Vol. 44, pp. 349-353 (1998).

[FSS] Furst, M. & Saxe, J. & Sipser, M. "Parity, Circuits, and the Polynomial-Time Hierarchy" Mathematical Systems Theory, Vol. 17, pp. 13-27 (1984).

[GLL] Grumbach, S. & Lacroix, Z. & Lindell, S. "Generalized Implicit Definitions on Finite Structures" CSL 1995: 252-265

[L] Libkin, L. *Elements of Finite Model Theory* Springer, 2004.

[M] Moschovakis, Y. *Elementary induction on abstract structures* North-Holland, 1974.

[I] Immerman, N. *Descriptive complexity* Springer, 1999.

A CATEGORICAL APPROACH TO CONSTRAINT SATISFACTION PROBLEM

JAKUB OPRŠAL

What distinguished problems that are NP-hard from those that allow a polynomial time algorithm? Arguably resolving this question would entail a proof that P is not NP. Several fields of theoretical computer science have (or had) an approach to answering it, but again and again it showed to be incredibly hard. I work in a field that approaches this question through studying complexity of a well-structured class of problems in NP—non-uniform *constraint satisfaction problems*. The structure involved in these problems allow us to formally connect attributes of the problems with their complexity.

The *constraint satisfaction problem (CSP)* is a prototypical NP-complete problem. Loosely speaking, the goal is given an instance consisting of variables, that attain values over some (usually finite) domain, and constraints, each involving a finite number of variables and given by the set of permissible tuples, decide whether there is an assignment of values to the variables that satisfies all the constraints. A focus of research in the past years has been to classify the complexity of this problem depending on the shape of the constraints allowed. This is better explained when the CSP is expressed as a homomorphism problem: Given two finite structures $\mathbf{A}$ and $\mathbf{B}$ in the same relational language, decide whether there is a homomorphism from $\mathbf{A}$ to $\mathbf{B}$. Fixing the structure $\mathbf{B}$, we obtain the problem CSP($\mathbf{B}$) whose complexity depends on properties of the structure $\mathbf{B}$, e.g., if $\mathbf{B}$ is the 3-clique, the problem is NP-hard, while if relations of $\mathbf{B}$ are defined by affine equations, the problem is in P.

Algebraic approach was the go to theory for approaching complexity classification of these problems, and it played a key role in Bulatov's and Zhuk's celebrated results [Bul17, Zhu20] that showed that, for any finite $\mathbf{B}$, CSP($\mathbf{B}$) is NP-hard or in P. The core result of this theory is a classification of when one such CSP can be reduced to another using *simple gadget reductions*. This characterisation uses the notion of *polymorphism*. One of advantages of polymorphisms in the realm of finite template CSPs is that they correctly identify the NP-hard templates—a finite template CSP is NP-hard if and only if the polymorphisms of its template do not satisfy any non-trivial algebraic (height 1) identity.

Recently, a generalisation of the CSP, so-called *promise constraint satisfaction problem (PCSP)* gained a lot of attention for a few good reasons: It still retains the structure of the CSP, it is more general, and includes problems of interest to wide audience. A template of a promise constraint satisfaction problem consists of two structures $\mathbf{A}$ and $\mathbf{B}$ such that $\mathbf{A}$ maps homomorphically to $\mathbf{B}$ (e.g., $\mathbf{A} = K_3$ and $\mathbf{B} = K_5$). This problem then asks,

Date: May 21, 2021.

given a third structure $\mathbf{C}$ that is promised to map homomorphically to $\mathbf{A}$, find a homomorphism from $\mathbf{C}$ to $\mathbf{B}$ (e.g., given a graph that is promised to be 3-colourable, find a 5-colouring of this graph).

The algebraic theory of CSPs can adapted to promise constraint satisfaction problems [BBKO19]. The notion of a polymorphism easily generalises. Nevertheless, it has a few shortcomings: Polymorphisms do not satisfy the same strong closure properties making many algebraic results unusable in this settings. Further, unlike for CSPs, the theory does not correctly identify the NP-hard problems. This means that in order to progress in classification of complexity of PCSPs, we need to refine our old tools, and discover new tools. Several recent papers [AGH17, DRS05, BG18, KO19, WŽ20] show that many various tools can be useful. Let me highlight two papers [KO19, WŽ20] (and their joined journal version [KOWŽ20]) that make surprising use abstracts methods from algebraic topology and ideas of category theory. In the talk, I will outline the basics of the algebraic theory from a new perspective that was brought by these papers.

Contributions. The essence of the algebraic approach to (P)CSP is classification when one CSP can be reduced to another using a *simple gadget reduction* by the means of some higher symmetries of the template, called *polymorphisms*. The shortcoming of this theory is that these reductions are simply not enough in the world of promises—we need new, more general reductions.

The new insight is that reductions in the CSP setting are given by (a weaker version of) adjunctions. For example, a simple gadget reduction has an associated *pp-power* construction Γ such that

$$\Lambda(\mathbf{A}) \to \mathbf{B} \text{ if and only if } \mathbf{A} \to \Gamma(\mathbf{B}).$$

(Here, we denote by $\mathbf{A} \to \mathbf{B}$ that a homomorphism from $\mathbf{A}$ to $\mathbf{B}$ exists.) If the above is true for some Λ and Γ, we will say that Λ and Γ are *thinly adjoint*. This property is actually enough to show that Λ is a reduction from $\mathrm{CSP}(\Gamma(\mathbf{B}))$ to $\mathrm{CSP}(\mathbf{B})$. In other words, if some functors Λ and Γ are thinly adjoint, and Λ is efficiently computable, then there is an efficient reduction from $\mathrm{CSP}(\Gamma(\mathbf{B}))$ to $\mathrm{CSP}(\mathbf{B})$.

Reductions comming from such thin adjoints can be also used in the PCSP setting: as one can easily check, if Λ and Γ are thinly adjoint, then Λ gives a reduction from $\mathrm{PCSP}(\mathbf{A}_1, \mathbf{B}_1)$ to $\mathrm{PCSP}(\mathbf{A}_2, \mathbf{B}_2)$ if $\Lambda(\mathbf{A}_1) \to \mathbf{A}_2$ and $\Gamma(\mathbf{B}_2) \to \mathbf{B}_1$. A new reduction of this form has been used by Wrochna and Živný [WŽ20] to provide new results on approximate graph colouring—they showed that for any $k \geq 4$, it is NP-hard to colour a k-colourable graph with $\binom{k}{\lfloor k/2 \rfloor} - 1$ colours.

The core of the algebraic approach can be compressed to the following theorem.

Theorem 1. *The following are equivalent for finite relational structures* $\mathbf{A}$*,* *and* $\mathbf{B}$*.*

(1) *There is a simple gadget reduction (and therefore a log-space reduction) from* $\mathrm{CSP}(\mathbf{A})$ *to* $\mathrm{CSP}(\mathbf{B})$*.*

(2) $\mathbf{A}$ *is homomorphically equivalent to a pp-power of* $\mathbf{B}$ *(i.e., there is a pp-power* $\Gamma(\mathbf{B})$ *of* $\mathbf{B}$ *s.t.* $\Gamma(\mathbf{B}) \to \mathbf{A}$ *and* $\mathbf{A} \to \Gamma(\mathbf{B})$*).*
(3) *There is a 'minion homomorphism' from* $\mathrm{pol}(\mathbf{B})$ *to* $\mathrm{pol}(\mathbf{A})$.

In the previous paragraph, we have already argued that (1) and (2) are equivalent—and indeed that is the easier part of the above theorem. The strength of this theorem lies in equivalence of these items with (3), and the hard part of the proof is to show that (3) implies either (1) or (2). I will present a new proof of this fact that takes advantage of the above observation about adjoints, and adjunction of three constructions: the *polymorphism minion* (as a functor from relational structures to minions), the *indicator instance*, and *minor conditions* constructed from a CSP instance. Examples of these constructions appeared many times in various proofs, but with the exception of polymorphisms they were rarely systematically studied.

Acknowledgements. This is a joint work with Andrei Krokhin, Marcin Wrochna, and Stanislav Živný.

References

[AGH17] Per Austrin, Venkatesan Guruswami, and Johan Håstad. $(2+\varepsilon)$-Sat is NP-hard. *SIAM J. Comput.*, 46(5):1554–1573, 2017. doi:10.1137/15M1006507.

[BBKO19] Libor Barto, Jakub Bulín, Andrei Krokhin, and Jakub Opršal. Algebraic approach to promise constraint satisfaction. June 2019. arXiv:1811.00970v3.

[BG18] Joshua Brakensiek and Venkatesan Guruswami. Promise constraint satisfaction: Structure theory and a symmetric boolean dichotomy. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA'18, pages 1782–1801, Philadelphia, PA, USA, 2018. Society for Industrial and Applied Mathematics. arXiv:1704.01937, doi:10.1137/1.9781611975031.117.

[Bul17] Andrei A. Bulatov. A dichotomy theorem for nonuniform CSPs. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 319–330, October 2017. doi:10.1109/FOCS.2017.37.

[DRS05] Irit Dinur, Oded Regev, and Clifford Smyth. The hardness of 3-uniform hypergraph coloring. *Combinatorica*, 25(5):519–535, September 2005. doi:10.1007/s00493-005-0032-4.

[KO19] Andrei Krokhin and Jakub Opršal. The complexity of 3-colouring H-colourable graphs. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS'19)*, pages 1227–1239, 2019. arXiv:1904.03214, doi:10.1109/FOCS.2019.00076.

[KOWŽ20] Andrei Krokhin, Jakub Opršal, Marcin Wrochna, and Stanislav Živný. Topology and adjunction in promise constraint satisfaction. March 2020. arXiv:2003.11351.

[WŽ20] Marcin Wrochna and Stanislav Živný. Improved hardness for H-colourings of G-colourable graphs. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'20)*, pages 1426–1435, 2020. arXiv:1907.00872, doi:10.1137/1.9781611975994.86.

[Zhu20] Dmitriy Zhuk. A proof of the CSP dichotomy conjecture. *J. ACM*, 67(5):30:1–30:78, August 2020. doi:10.1145/3402029.

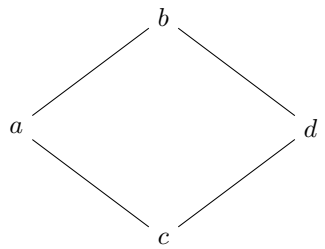
Email address: jakub.oprsal@durham.ac.uk

Department of Computer Science, Durham University, Durham, UK

GRAPH TRAVERSALS AS UNIVERSAL CONSTRUCTIONS (EXTENDED ABSTRACT)

SIDDHARTH BHASKAR*

Problem statement. Graph searches are algorithms for visiting the vertices in a connected graph from a prescribed source. Both graph searches and their resulting vertex orders, or *traversals*, are absolutely fundamental in the theory of graph algorithms, and have important applications in other areas of theoretical computer science such as computational complexity theory.



We start from the premise that a concept as natural as a traversal should be canonical in the original graph. For example, let us consider the graph to the left, and fix a as a source. Of the six vertex orderings starting with a, (a, d, b, c) and (a, d, c, b) are not traversals: each vertex added during a search must be in the boundary of previously visited vertices, but d is not in the boundary of $\{a\}$.

Of the four remaining vertex orders, two are breadth-first traversals ((a, b, c, d) and (a, c, b, d)) and two are depth-first traversals ((a, b, d, c) and (a, c, d, b)). This can easily be checked by hand: in a breadth-first search, we go level-by-level, and must visit both b and c before we visit d. In a depth-first search, we prioritize the neighbor of the most recently visited vertex, so we visit d before the latter of $\{b, c\}$.

However, notice that there is no way of canonically distinguishing between the two breadth-first or the two depth-first traversals. Concretely, once we visit a, there is no canonical way to choose between b and c. A natural fix is to linearly order each neighborhood and visit lesser neighbors first. We call the resulting traversals *lexicographic*. For example, if we say that $b < c$, then the lexicographic breadth-first traversal is (a, b, c, d) and the lexicographic depth-first traversal is (a, b, d, c). If we say that $c < b$, we get (a, c, b, d) and (a, c, d, b) respectively. We call a graph whose neighborhoods are linearly ordered an *edge-ordered* graph.

Our contribution. We show that both the lexicographic breadth-first traversal and lexicographic depth-first traversal are canonically obtainable from a given finite, edge-ordered graph with a distinguished source. Specifically, we equip edge-ordered graphs with two different kinds of monotone graph homomorphisms, one preserving lexicographically least paths and the other lexicographically least *shortest* paths, to obtain two categories: $\mathbf{FinGraph}^{l}_{\star}$ and $\mathbf{FinGraph}^{s}_{\star}$.

We obtain the lexicographic depth-first traversal via a functor out each category and we furthermore factor each functor as sequence of *universal* (free and cofree) constructions on edge-ordered graphs, plus a single forgetful functor that takes an edge ordered graph and simply extracts the ordered neighborhood of the distinguished source.

At a first approximation, each lexicographic traversal can be expressed as the composition of a least-path tree functor plus some sort of transitive closure into a category of transitive ordered graphs, as indicated below. (The category $\mathbf{FinArb}^{<}_{\star}$ is the category of *finite edge-ordered arborescences*. See also

*This work is joint with Robin Kaarsgaard.

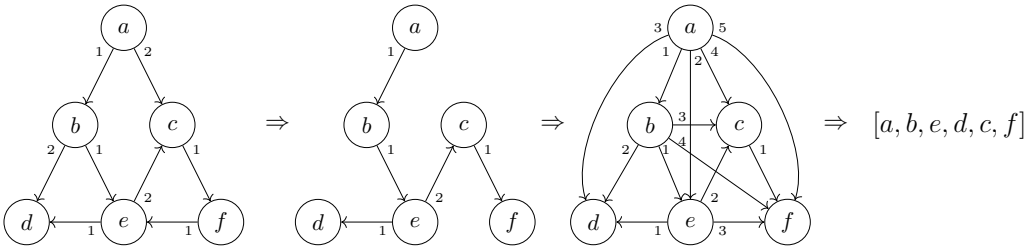


FIGURE 1. The construction of the lexicographic depth-first traversal starting from a on a given edge-ordered graph. First we extract the least-path tree, transitively close it, then isolate the ordered neighborhood of a. Numerals indicate the edge ordering. Each transformation is universal, except for a "silent" (forgetful) transformation fixing the transitive graph but forgetting some structure on morphisms.

Figure 1 for an example on a concrete graph.) This decomposition was first observed in [3]—not in the context of category theory— and used to derive efficient parallel algorithms; see also [2, 4].

$$\mathbf{FinGraph}^{l}_{\star} \;\top\; \mathbf{FinArb}^{<}_{\star} \;\bot\; \mathbf{TLexGraph} \qquad \mathbf{FinGraph}^{s}_{\star} \;\top\; \mathbf{FinArb}^{<}_{\star} \;\bot\; \mathbf{TArb}$$

The main technical problem we solve is identifying precisely the right notions of edge-ordered graphs and homomorphisms that allow us to formulate least-path trees and transitive closures as universal constructions. For example, there is a well-known adjunction between graphs and categories, essentially by transitively closing the edge relation, but it is not clear how to lift this into the edge-ordered setting so that the resulting operation is functorial, let alone universal.

Structure meets power. A basic question in computer science is that of the relationship between an *algorithm* and its *implementations* [9]. We view this problem as one aspect of the "structure vs. power" dichotomy of Abramsky [1]. At the risk of oversimplification, one understands algorithms structurally: a standard taxonomy of algorithms arises from closing a core set of basic paradigms (iteration, divide-and-conquer, a greedy strategy, etc.) under basic compositional operations (sequential composition, subroutines, etc.). On the other hand, one typically measures expressive power with respect to a concrete implementation.

Building a robust network of links between algorithms and their implementations is one realizations of Abramsky's vision. We can imagine a world, for example, where formally describing the structure of some algorithm might immediately yield some information about its possible machine implementations, its parallel complexity, its sequential complexity, and even lower bounds. We view our work as a small first step along this road for some very fundamental algorithms, viz. breadth- and depth-first graph traversals.

To the best of our knowledge, equipping algorithmic problems with a categorical structure is a relatively recent idea. While graphs have been studied extensively from a categorical point of view, the focus has been on topics such as graph rewriting and string diagrams [5, 7] and relationships with properads [6] rather than graph algorithms. The only other papers that we know of in the algorithmic vein are that of Master on the open algebraic path problem [8] and the compositional algorithm by Rathke, Sobociński and Stephens [10] for reachability in Petri nets. These papers suggest that there might be a categorical setting for reachability problems on graphs, generally understood.

In the medium-to-long term, we envision developing a theory of *compositional graph algorithms through universal properties* as a step along this way. A goal of such a project would be "reverse implementation," namely, canonically recovering implementations from algorithms, if the latter are suitably formulated. Concretely, can we use our categorical decompositions of breadth- and depth-first traversals to canonically arrive at their sequential implementations using *queues* and *stacks*?

References

[1] S. Abramsky. Whither semantics? *Theoretical Computer Science*, 807:3–14, 2020.
[2] E. Allender, A. Chauhan, and S. Datta. Depth-first search in directed graphs, revisited. *Electronic colloquium on computational complexity*, 20(74), 2020.
[3] P. Delatorre and C.P. Kruskal. Fast parallel algorithms for all-sources lexicographic search and path-algebra problems. *Journal of Algorithms*, 19(1):1–24, 1995.
[4] P. Delatorre and C.P. Kruskal. Polynomially improved efficiency for fast parallel single-source lexicographic depth-first search, breadth-first search, and topological-first search. *Theory of Computing Systems*, 34:275–298, 2001.
[5] L. Dixon and A. Kissinger. Open-graphs and monoidal theories. *Mathematical Structures in Computer Science*, 23(2):308–359, 2013.
[6] J. Kock. Graphs, hypergraphs, and properads. *Collectanea mathematica*, 67(2):155–190, 2016.
[7] S. Lack and P. Sobociński. Adhesive and quasiadhesive categories. *RAIRO-Theoretical Informatics and Applications*, 39(3):511–545, 2005.
[8] J. Master. The open algebraic path problem, 2020. `arXiv:arXiv:2005.06682`.
[9] Yiannis N. Moschovakis and Vasilis Paschalis. *Elementary Algorithms and Their Implementations*, pages 87–118. Springer New York, New York, NY, 2008.
[10] J. Rathke, P. Sobociński, and O. Stephens. Compositional reachability in petri nets. In J. Ouaknine, I. Potapov, and J. Worrell, editors, *Reachability Problems*, pages 230–243. Springer, 2014.

University of Copenhagen, Denmark
Email address: `sbhaskar@di.ku.dk`

On games on algebras

Glynn Winskel

There are several reasons to integrate finite model theory (FMT) within a general theory of games. Historically two-party games (with Player versus Opponent, or Duplicator versus Spoiler) have provided central techniques of FMT. The operational nature of games brings them close to resources whose control and use are the concerns of FMT. However to go beyond the games traditionally used in FMT requires games and strategies which both support structure and resource. In the semantic world, concurrent games and strategies based on event structures have been demonstrated to do both: concurrent games and strategies can be composed and support quantitative extensions, *e.g.* to probabilistic and quantum computation. In short, concurrent games and strategies deserve study as a potentially rich arena in which structure meets power.

In this talk I will present concurrent games and strategies over algebras with many-sorted signature Σ. (The use of many sorts allows individual games and strategies to involve several algebras.) Through their foundation in event-structures, concurrent games and strategies represent closely the operational nature of games, their interactivity, the dependence, independence and conflict of moves.

A Σ-game will be represented by an event structure in which each event (standing for a move occurrence) both carries

- a polarity to signify whether it represents a move of Player (Duplicator) or Opponent (Spoiler);
- and a variable drawn from a set Var of Σ-sorted variables, in such a way that no two concurrent events are assigned the same variable.

Positions of the game are represented by configurations of the event structure. The condition on the assignment of variables ensures that events with the same variable are totally ordered within a configuration; so it makes sense to talk of the "latest" move associated with a variable appearing within a finite configuration. Winning configurations of the game are specified by an assertion in the free logic over Σ w.r.t. the latest instantiations of the variables. (By using a free logic we avoid clumsy indexing by the configurations to which assertions apply.)

With respect to a Σ-algebra $\mathcal{A}$, an $\mathcal{A}$-strategy for Player is a concurrent strategy which roughly makes choices of values in $\mathcal{A}$ for Player variables (those associated with Player moves) in response to choices of values for Opponent variables. It is winning if it results in a winning configuration of the game no matter what the strategy of Opponent. Traditional pebbling games such

as homomorphism games and the isomorphism games of Ehrenfeucht-Fraissé provide examples of such games and strategies.

However, one advantage of concurrent games and strategies is that they support composition to form a bicategory, and a category in the case of deterministic strategies. The bicategory is constructed by following the ideas of Conway and Joyal: a strategy *from* a game G *to* a game H is a strategy in the compound game $G^{\perp}||H$, formed as the dual game of G, in which the roles of Player and Opponent are reversed, in parallel with H. Then the composition of a strategy from G to H and a strategy from H to K is obtained by playing the strategies against each other over the common game H. Another advantage of concurrent games is that they also support imperfect information, essentially by restricting strategies to those causal dependencies allowed by an accessibility preorder. Assumptions of imperfect information often play a role in reasoning about the power of additional resources, for example in games to show quantum advantage.

Making use of these advantages, we can provide full and faithful embeddings into the category of deterministic strategies of the categories of Σ-algebras with homomorphisms as well as coKleisli maps with respect to pebbling comonads. (The results are subject to minor restrictions on the algebras.) This reconciles the composition of strategies as coKleisli maps, prevalent in recent algebraicisation of FMT, with the more usual composition of strategies, following the paradigm of Conway and Joyal.