Treewidth via Spined Categories (extended abstract)

Zoltan A. Kocsis
Commonwealth Scientific and Industrial Research Organisation
Eveleigh NSW, Australia
zoltan.kocsis@csiro.au

Benjamin Merlin Bumpus
University of Glasgow
Scotland, UK
benjamin.merlin.bumpus@gmail.com

Treewidth is a well-known graph invariant with multiple interesting applications in combinatorics. On the practical side, many NP-complete problems are polynomial-time (sometimes even linear-time) solvable on graphs of bounded treewidth [4, 5]. On the theoretical side, treewidth played an essential role in the proof of the celebrated Robertson-Seymour graph minor theorem [10]. While defining treewidth-like invariants on graphs [3, 6, 9] and treewidth analogues on other sorts of combinatorial objects (incl. hypergraphs, digraphs [7, 8]) has been a fruitful avenue of research, a direct, categorial description capturing multiple treewidth-like invariants is yet to emerge.

Here we report on our recent work on *spined categories* [2]: categories equipped with extra structure that permits the definition of a functorial analogue of treewidth, the *triangulation functor*. The usual notion of treewidth is recovered as a special case, the triangulation functor of a spined category with graphs as objects and graph monomorphisms as arrows. The usual notion of treewidth for hypergraphs arises as the triangulation functor of a similar category of hypergraphs.

1 Spined Categories

Contrary to the usual convention in category-theoretic texts, we use the word *graph* to refer to simple graphs (irreflexive, without loops or multiedges). We write Grph for the category that has graphs as objects and graph homomorphisms as arrows, and Grph_m for the category with the same objects, but monomorphisms as arrows.

Definition 1.1. A *spined category* consists of a category $\mathcal{C}$ equipped with the following additional structure:

- a sequence $\Omega : \mathbb{N} \to \mathrm{ob}\,\mathcal{C}$ called the *spine* of $\mathcal{C}$,
- an operation $\mathfrak{P}$ (called the *proxy pushout*) that assigns to each diagram of the form $G \xleftarrow{g} \Omega_n \xrightarrow{h} H$ in $\mathcal{C}$ a distinguished cocone $G \xrightarrow{\mathfrak{P}(g,h)_g} \mathfrak{P}(g,h) \xleftarrow{\mathfrak{P}(g,h)_h} H$,

subject to the following two conditions:

SC1 For every $X \in \mathrm{ob}\,\mathcal{C}$ there is $n \in \mathbb{N}$ such that $\mathcal{C}(X, \Omega_n) \neq \emptyset$.

SC2 Given any diagram of the form $G' \xleftarrow{g'} G \xleftarrow{g} \Omega_n \xrightarrow{h} H \xrightarrow{h'} H'$ we can find a *unique* morphism $(g', h') : \mathfrak{P}(g,h) \to \mathfrak{P}(g' \circ g, h' \circ h)$ making the following diagram commute:

$$\begin{array}{ccccc}
\Omega_n & \xrightarrow{g} & G & \xrightarrow{g'} & G' \\
{\scriptstyle h}\downarrow & & \downarrow{\scriptstyle \mathfrak{P}(g,h)_g} & & \\
H & \xrightarrow[\mathfrak{P}(g,h)_h]{} & \mathfrak{P}(g,h) & & \downarrow{\scriptstyle \mathfrak{P}(g'\circ g, h'\circ h)_{g'\circ g}} \\
{\scriptstyle h'}\downarrow & & & \searrow{\scriptstyle (g',h')} & \\
H' & & \xrightarrow[\mathfrak{P}(g'\circ g, h'\circ h)_{h'\circ h}]{} & & \mathfrak{P}(g'\circ g, h'\circ h)
\end{array}$$

Proxy pushouts capture an important property that the left-cancellative subcategory Grph_m "remembers" about the existence of pushouts in the category Grph: we can equip the former with proxy pushouts by assigning to each diagram $G \xhookleftarrow{g} \Omega_n \xhookrightarrow{h} H$ its pushout square in the latter. Moreover, a category $\mathcal{C}$ with all pushouts, when equipped with a sequence $\Omega : \mathbb{N} \to \mathrm{ob}\,\mathcal{C}$ satisfying **SC1**, always forms a spined category.

Submitted to:
ACT 2021

Definition 1.2. Consider spined categories $(C, \Omega^C, \mathfrak{P}^C)$ and $(D, \Omega^D, \mathfrak{P}^D)$. We call a functor $F : C \to D$ a *spinal functor* if it

SF1 *preserves the spine*, i.e. $F \circ \Omega^C = \Omega^D$, and

SF2 *preserves proxy pushouts*, i.e. given a proxy pushout square

$$\begin{array}{ccc} \Omega_n & \xrightarrow{g} & G \\ {\scriptstyle h}\downarrow & & \downarrow{\scriptstyle \mathfrak{P}^C(g,h)_g} \\ H & \xrightarrow[\mathfrak{P}^C(g,h)_h]{} & \mathfrak{P}^C(g,h) \end{array}$$

in the category C, its F-image

$$\begin{array}{ccc} \Omega_n & \xrightarrow{Fg} & F[G] \\ {\scriptstyle Fh}\downarrow & & \downarrow{\scriptstyle F\mathfrak{P}^C(g,h)_g} \\ F[H] & \xrightarrow[F\mathfrak{P}^C(g,h)_h]{} & F[\mathfrak{P}^C(g,h)] \end{array}$$

forms a proxy pushout square in D. One can state this equationally, by demanding that the equalities $F[\mathfrak{P}^C(g,h)] = \mathfrak{P}^D(Fg, Fh)$, $F\mathfrak{P}^C(g,h)_g = \mathfrak{P}^D(Fg, Fh)_{Fg}$ and $F\mathfrak{P}^C(g,h)_h = \mathfrak{P}^D(Fg, Fh)_{Fh}$ all hold.

That is, a spinal functor between spined categories is a functor between the underlying categories that respects the spine and proxy pushout structure. As expected, the composition of two spinal functors is itself spinal.

Regard the poset $(\mathbb{N}, \leq)$ of natural numbers under the usual ordering as a category. This category has all pushouts. Equipping $(\mathbb{N}, \leq)$ with the spine $\Omega_n = n$ and suprema as proxy pushouts yields a simple example of a spined category, which we will denote Nat.

Definition 1.3. An *S-functor* on the spined category C is a spinal functor defined on C and valued in Nat.

Some spined categories do not have any S-functors defined on them: typically when some object Ω_n can be constructed as a proxy pushout using Ω_i for $i < n$. The interested reader is welcome to enumerate necessary/sufficient conditions for the existence of S-functors. In what follows, we side-step this issue by focusing our attention on the class of spined categories which have at least one S-functor defined on them. We call such categories *measurable*.

Example 1.4. The category Grph_m (with proxy pushouts inherited from pushouts in Grph, and spine Ω_n the complete[1] graph on n vertices) is measurable: it's easy to check that the map $\omega(G)$ which sends each G to the size of its largest complete subgraph, constitutes an S-functor.

2 Triangulation Functor

Our main result proves the existence of a distinguished S-functor, the *triangulation functor* on each measurable spined category. Treewidth is recovered as the triangulation functor of the category Grph_m, while hypergraph treewidth is recovered as the triangulation functor of a corresponding category HGrph_m. For traditional graph-theoretic definitions of treewidth, we refer the reader to Encyclopedia of Algorithms [1]: our *pseudo-chordal* objects play a similar role to that of chordal[2] graphs in the second characterisation presented there.

Definition 2.1. We call an object X of a spined category C *pseudo-chordal* if all S-functors assign the same value to X, i.e. for any two S-functors $F, G : C \to \mathrm{Nat}$ we have $F[X] = G[X]$. We let $\mathrm{pc}\, C$ denote the class of pseudo-chordal objects in the category C.

In the category Grph_m defined above, the class of pseudo-chordal objects forms a strict superset of the class of chordal graphs: while all chordal graphs are in fact pseudo-chordal objects, the converse fails.

[1] A graph where every pair of distinct vertices is connected by an edge.
[2] A graph where all cycles of > 3 vertices have a *chord*, i.e. an edge connecting non-adjacent vertices of the cycle.

Theorem 2.2 (Main result)**.** *Take a measurable spined category $\mathcal{C}$, equipped with some S-functor $s : \mathcal{C} \to \text{Nat}$. The map $\Delta : \mathcal{C} \to \text{Nat}$ defined by the equation $\Delta[G] = \min\left\{ s(H) \,\middle|\, H \in \text{pc}\,\mathcal{C}, \mathcal{C}(G,H) \neq \emptyset \right\}$*

1. *is an S-functor;*
2. *dominates all other S-functors, i.e. for any $X \in \text{ob}\,\mathcal{C}$ and S-functor $F : \mathcal{C} \to \text{Nat}$, $F[X] \leq \Delta[X]$.*

We call the functor Δ the triangulation functor of the category $\mathcal{C}$. It's clear that every measurable category has a unique triangulation functor.

Theorem 2.3. *The triangulation functor of the category* Grph_m *coincides with treewidth.*

Spined categories socialize well via spinal functors: in the talk, we will explain how one can obtain measurability (and non-measurability) results purely by constructing spinal functors, and present further examples, including a category of hypergraphs where the triangulation functor recovers the notion of hypergraph treewidth. Some previously unknown tree-width-like invariants also emerge by collecting the relevant combinatorial objects into a spined category. Somewhat surprisingly, by putting mild computability conditions on the category, we can even obtain an algorithm which computes the value of the triangulation functor (although the generic algorithms obtained this way are impractically slow for computing the treewidth of all but the simplest graphs).

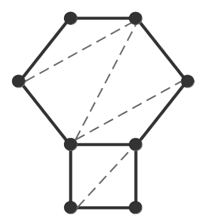

Figure 1: A graph and one of its chordal completions (dashed). The treewidth of a graph G is the infimum of the sizes of the largest complete subgraphs contained in the chordal completions of G.

References

[1] H. L. Bodlaender (2016): *Treewidth of Graphs*, pp. 2255–2257. Springer New York, New York, NY, doi:https://doi.org/10.1007/978-1-4939-2864-4_431.

[2] B. M. Bumpus & Z. A. Kocsis (2021): *Spined categories: generalizing tree-width beyond graphs*. *arXiv e-prints*:arXiv:2104.01841, submitted to Combinatorica.

[3] B. Courcelle, J. Engelfriet & G. Rozenberg (1993): *Handle-rewriting hypergraph grammars*. *Journal of Computer and System Sciences* 46(2), pp. 218–270, doi:https://doi.org/10.1016/0022-0000(93)90004-G.

[4] M. Cygan, F. V. Fomin, Ł. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk & S. Saurabh (2015): *Parameterized algorithms*. Springer, doi:https://doi.org/10.1007/978-3-319-21275-3.

[5] J. Flum & M. Grohe (2006): *Parameterized Complexity Theory. 2006*. *Texts Theoret. Comput. Sci. EATCS Ser*, doi:https://doi.org/10.1007/3-540-29953-X.

[6] M. Habib & C. Paul (2010): *A survey of the algorithmic aspects of modular decomposition*. *Computer Science Review* 4(1), pp. 41 – 59, doi:https://doi.org/10.1016/j.cosrev.2010.01.001.

[7] T. Johnson, N. Robertson, P. D. Seymour & R. Thomas (2001): *Directed tree-width*. *Journal of Combinatorial Theory. Series B* 82(1), pp. 138–154, doi:https://doi.org/10.1006/jctb.2000.2031.

[8] S. Kreutzer & O.-j. Kwon (2018): *Digraphs of Bounded Width*. In: *Classes of Directed Graphs*, Springer, pp. 405–466, doi:https://doi.org/10.1007/978-3-319-71840-8_9.

[9] N. Robertson & P. D. Seymour (1991): *Graph minors X. Obstructions to tree-decomposition*. *Journal of Combinatorial Theory, Series B* 52(2), pp. 153–190, doi:https://doi.org/10.1016/0095-8956(91)90061-N.

[10] N. Robertson & P.D. Seymour (2004): *Graph Minors. XX. Wagner's conjecture*. *Journal of Combinatorial Theory, Series B* 92(2), pp. 325 – 357, doi:https://doi.org/10.1016/j.jctb.2004.08.001.