

Replace this file with `prentcsmacro.sty` for your meeting,  
or with `entcsmacro.sty` for your meeting. Both can be  
found at the [ENTCS Macro Home Page](#).

# Towards Compositional Graph Theory

Apiwat Chantawibul and Paweł Sobociński

*ECS, University of Southampton*

*Dedicated to the memory of R.F.C. Walters.*

---

## Abstract

Decomposing graphs into simpler graphs is one of the central concerns of graph theory. Investigations have revealed deep concepts such as *modular decomposition*, *tree width* or *rank width*, which measure—in different ways—the *structural complexity* of a graph's topology. Courcelle and others have shown that such concepts can be used to obtain efficient algorithms for families of graphs that are amenable to decomposition (e.g. those that have bounded tree-width). These algorithms, in turn, are of course of use in computer science, where graphs are ubiquitous. In this paper we take the first steps towards understanding notions of decomposition in graph theory *compositionally*, and more generally, in a categorical setting: category theory, after all, is the mathematics of compositionality. We introduce the concept of  $\cup$ -*matrices* (cup-matrices). Like ordinary matrices,  $\cup$ -matrices are the arrows of a PROP: we give a presentation, extending the work of Lafont, and Bonchi, Zanasi and the second author. A variant of  $\cup$ -matrices is then used in the development of a novel algebra of simple graphs, the *lingua franca* of graph theory. The algebra is that of a certain symmetric monoidal theory:  $\cup$ -matrices—akin to adjacency matrices—encode the graphs' topology.

---

## 1 Introduction

When category theorists talk about the category of graphs, they usually mean the presheaf category  $\mathbf{Graph} = \mathbf{Set}^{\Rightarrow}$ . The objects of this category, however, are not what graph theorists would typically refer to as *graphs* tout court – rather, as *directed multigraphs*, because there may be more than one directed edge between any two vertices. The most basic—and important—objects of study in graph theory are *simple graphs*: these are undirected with at most one edge between any two vertices, and no self loops. In this paper, we are concerned with simple graphs.

A *structural metric* of a graph is a way of assigning a numerical value, typically a natural number, to a graph. The intention is for the number to indicate, in some way, the graph's inherent *structural complexity*. Some well-known structural metrics in graph theory include path-width, tree-width, branch-width, clique-width and rank-width. The notion of tree-width is perhaps the best known in Computer Science through the work of Courcelle [6] who showed that monadic second-order logic can be decided in linear time for families of graphs with *bounded* tree-width. Courcelle's theorem has found several algorithmic applications (see e.g. [13]).

The structural metric *rank-width*, due to Oum and Seymour, has been a hot topic in graph theory over the last ten years and is of particular relevance for us. We refer to [17, 18] for the technical details, here we give an intuitive description. A *rank-decomposition* of a simple graph with vertex set  $V$  can be considered as a binary tree, where the tree-nodes are labelled with nonempty subsets of  $V$  and the tree-edges are labelled with natural numbers, such that:

- the labels (vertex sets  $W_1, W_2$ ) of the children  $w_1, w_2$  of any tree-node  $w$  are a (binary) partition of the label (vertex set  $W$ ) of the parent tree-node: i.e.  $W = W_1 \cup W_2$  and  $W_1 \cap W_2 = \emptyset$ ,
- the root is labelled with  $V$ ,
- the leaves are labelled with singletons,
- the tree-edge from a parent to a child labelled with vertex set  $W$  is labelled with the *rank* of the  $|V \setminus W| \times |W|$  adjacency  $\mathbb{Z}_2$ -matrix that tabulates the edges from  $W$  to the remainder of the graph. Note that matrix algebra is performed over the field  $\mathbb{Z}_2$ , i.e.  $1 + 1 = 0$ .

The *width* of a particular rank-decomposition is its maximum edge label. The *rank-width* of a graph is then the width of an optimal rank-decomposition: one with the smallest width. Discrete graphs and cliques both enjoy a rank-width of 1.

The concepts of rank-width and other structural metrics have proved to be very important in graph theory and related areas. There are two shortcomings, however, where category theory can help:

- *Generality.* The definition, as stated, is specialised to simple graphs. Yet, the underlying concept is quite robust and can be stated *mutatis mutandis* for other kinds of graphical structures: multigraphs, directed graphs (bi-rank-width), hypergraphs, Petri nets etc. This suggests that the fundamental theory ought to be done in a more general setting.
- *Compositionality.* The notion of a rank-decomposition (and equivalent notions for other structural metrics) is not inherently *compositional* in the sense that knowing how to decompose a graph  $G$  may not help in constructing decompositions of a graph  $H$  that has  $G$  as a sub-component. Intuitively speaking, rank-decompositions forget too much: only rank is recorded – the adjacency information ought to be recorded as well, in some form.

What, then, is a fruitful way of treating simple graphs categorically, in a way that will lead us to understanding structural metrics generally and compositionally? Graphs as objects and homomorphisms as arrows is the traditional approach, yet it does not immediately yield an *algebra* of graphs: for that, we need graphs to be the arrows. Cospans are a standard technique for turning objects into arrows, and the algebra of cospans of directed graphs was considered in [9], which is close in spirit to our work, although we do not consider cospans.

Similarly to Fiore and Campos’ work on an algebra of directed acyclic graphs [8], our approach is to use *symmetric monoidal theories*, by which we mean presentations of PROPs with generators and equations. The paper culminates with the symmetric monoidal theory ABUV, which we consider as an algebra of simple graphs. The role of matrix algebra is central: we build on Lafont’s [15] and Bonchi, Zanasi and the

second author’s [3] work on presentations of the PROP of matrices.

In order to understand the semantic counterpart of ABUV, we introduce the concept of  $\cup$ -matrices (“cup” matrices) that generalise matrix algebra and somewhat relax the straightjacket of directionality in ordinary matrices. The main technical contribution in this paper is a presentation of  $\cup$ -matrices. We conclude with some remarks on a structural metric, *monoidal width*, for symmetric monoidal theories.

The use of the algebra of monoidal categories to model systems of various kinds was pioneered by R.F.C. Walters and collaborators [11, 12]. Although we do not concentrate on applications in this paper, we believe that our theory will be relevant in several settings as there has been a recent surge in the applications of symmetric monoidal theories: amongst other works we mention signal flow graphs [1, 4], Petri nets [19–21], asynchronous circuits [10] and quantum circuits [5, 7].

**Structure of the paper.**

In §2 we recall the presentation of the PROP of matrices. In §3 we introduce  $\cup$ -matrices and develop their algebra. In §4 we give a presentation, and in §5 we use the earlier technical developments to introduce an algebra of simple graphs.

**Notation and background.**

Given an  $m_1 \times n_1$  matrix  $A$  and  $m_2 \times n_2$  matrix  $B$ , we write  $A \oplus B$  for the  $(m_1 + m_2) \times (n_1 + n_2)$  matrix  $\begin{pmatrix} A & 0 \\ 0 & B \end{pmatrix}$ . If  $A$  is an  $m \times n$  matrix,  $A^T$  denotes its transpose, an  $n \times m$  matrix, where  $A_{ij}^T = A_{ji}$ . A matrix  $A$  is upper triangular when it has only 0 entries below the main diagonal, i.e.  $A_{ij} = 0$  if  $i > j$ .

A PROP [14, 16] is a symmetric strict monoidal category with natural numbers as objects and with addition as the monoidal product on objects:  $m \oplus n = m + n$ . PROP homomorphisms are identity-on-objects strict symmetric monoidal functors: that is they preserve objects, composition, monoidal products and symmetries on the nose. We reserve the term *symmetric monoidal theory* for PROPs that are freely generated from a *presentation*: a pair  $(\Sigma, E)$  where  $\Sigma$  is the set of *generators* and  $E$  is a set of *equations* (some authors say *relations*).

## 2 Matrices as a Symmetric Monoidal Theory

The results in this section can be found in [3], based on the work of [15].

Fix a ring  $R$  (e.g. the integers); this is the source for the entries of our matrices, which we will not explicitly reference during the development in order to reduce the number of subscripts. We use  $r, s$  to range over  $R$ .

**Definition 2.1** *The PROP  $\mathbf{Mat}$  has  $m \times n$  matrices as arrows  $n \rightarrow m$ ; composition is matrix multiplication  $B \circ A = BA$ . The symmetries are permutation matrices.*

**Definition 2.2 (HA)** *The symmetric monoidal theory  $\mathbf{HA}$  is the free PROP on generators  $\{\Delta : 1 \rightarrow 2, \perp : 1 \rightarrow 0, \nabla : 2 \rightarrow 1, \top : 0 \rightarrow 1\} \cup \{r : 1 \rightarrow 1\}_{r \in R}$  and equations listed in Fig. 1. In string diagrams the generators are rendered:*

$$\Delta = \text{---} \bullet \text{---} \begin{array}{c} \frown \\ \cup \end{array} \quad \perp = \text{---} \bullet \quad \nabla = \begin{array}{c} \smile \\ \cap \end{array} \bullet \text{---} \quad \top = \bullet \text{---} \quad r = \text{---} \boxed{r} \text{---}$$

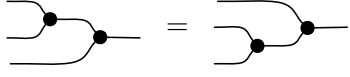
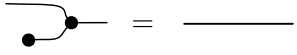
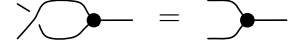
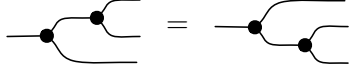
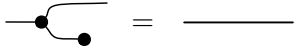
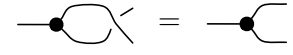
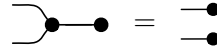
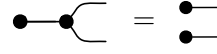
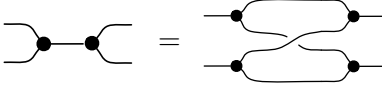

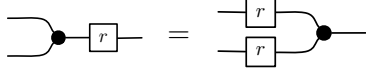
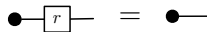
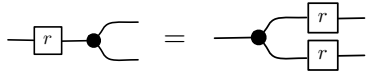

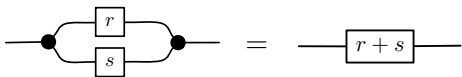
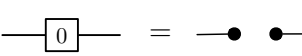
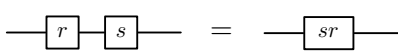
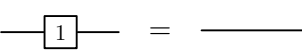
	$\nabla \circ (\nabla \oplus id) = \nabla \circ (id \oplus \nabla)$	(A1)
	$\nabla \circ (id \oplus \top) = id$	(A2)
	$\nabla \circ tw = \nabla$	(A3)
	$(\Delta \oplus id) \circ \Delta = (id \oplus \Delta) \circ \Delta$	(A4)
	$(id \oplus \perp) \circ \Delta = id$	(A5)
	$tw \circ \Delta = \Delta$	(A6)
	$\perp \circ \nabla = \perp \oplus \perp$	(A7)
	$\Delta \circ \top = \top \oplus \top$	(A8)
	$\Delta \circ \nabla = (\nabla \oplus \nabla) \circ (id \oplus tw \oplus id) \circ (\Delta \oplus \Delta)$	(A9)
	$\perp \circ \top = id_0$	(A10)
	$r \circ \nabla = \nabla \circ (r \oplus r)$	(A11)
	$r \circ \top = \top$	(A12)
	$\Delta \circ r = (r \oplus r) \circ \Delta$	(A13)
	$\perp \circ r = \perp$	(A14)
	$\nabla \circ (r \oplus s) \circ \Delta = r + s$	(A15)
	$0 = \perp \circ \top$	(A16)
	$s \circ r = sr$	(A17)
	$1 = id$	(A18)

Fig. 1. Equations of HA.

**Theorem 2.3** ([3, 15])  $HA \cong \text{Mat}$ .

□

### 3 $\cup$ -Matrices

Matrices are common in graph theory: for example, they are a convenient tool to record the connectivity information within a graph (adjacency matrices). They enjoy a rich and widely used algebra on which several techniques and results rely.

In our categorical algebra of graphs, introduced in §5, an arrow  $\mathbf{G} : m \rightarrow n$  combines three elements:

- a simple graph  $G$  on  $k$  vertices,
- a  $m \times k$  matrix,
- a  $n \times k$  matrix.

The idea is that the two matrices tell us how the vertices of  $G$  connect with other graphs that may be attached, at either end, through composition. Since  $G$  can itself be represented as a  $k \times k$  adjacency matrix, we are dealing with three matrices. We will see that relaxing the “directed” nature of matrices allows us to roll the three matrices into one  $\cup$ -matrix (pronounced “cup”-matrix).

#### 3.1 Turns

Before introducing  $\cup$ -matrices, we first need one important component, the notion of *turn* – an equivalence class of  $n \times n$  matrices that is closely connected with the concept of adjacency matrix in graph theory: two matrices are turn equivalent exactly when they encode the same connectivity information (in an undirected setting).

**Definition 3.1 (Turn equivalence, Turn)** *We say that two square  $k \times k$  matrices  $K$  and  $L$  are turn-equivalent when  $K + K^T = L + L^T$ , and write  $K \bowtie L$ . Clearly  $\bowtie$  is an equivalence relation. Note that if  $K \bowtie L$  then  $K$  and  $L$  have equal size.*

*By a  $(k)$ -turn we mean a  $\bowtie$  equivalence class of some  $k \times k$  matrix. We use  $M, N$  to range over turns, and given square  $K$ , we denote the induced turn by  $[K]$ .*

In certain cases, a turn has an obvious representative: an upper triangular matrix.

**Lemma 3.2** *Given a turn  $[K]$ , there exists an upper triangular matrix  $K'$  such that  $[K] = [K']$ . Suppose also that  $\forall r, s \in R$ , we have  $2r = 2s$  implies  $r = s$ <sup>1</sup>. If  $K$  and  $L$  are upper triangular square matrices, and  $[K] = [L]$ , then  $K = L$ .*

**Proof.** The first claim follows easily from the fact that addition in  $R$  is an abelian group. Next, if  $[K] = [L]$  then  $K + K^T = L + L^T$ . Since  $K$  and  $L$  are upper triangular, they clearly agree in all non diagonal entries. The  $i$ th diagonal entry of  $K + K^T = L + L^T$  is  $2K_{ii} = 2L_{ii}$ . Using the additional assumption on  $R$ ,  $K_{ii} = L_{ii}$ .  $\square$

The additional assumption on  $R$  in Lemma 3.2 excludes the case  $R = \mathbb{Z}_2$  where we have  $[(1)] = [(0)]$  so the second statement does not hold. In fact, the case of  $\mathbb{Z}_2$  is of particular importance in for our applications, and we will return to it in §5.

#### 3.2 $\cup$ -Matrices

A  $m \times n$   $\cup$ -matrix consists of an ordinary  $m \times n$  matrix together with an  $n$ -turn.

<sup>1</sup> This is the case, for instance, for the integers, or any field of characteristic not equal to 2

**Definition 3.3 (U-matrix)** An  $m \times n$  U-matrix  $U$  is a pair  $(A, M)$  where  $A$  is a  $m \times n$  matrix and  $M$  is an  $n$ -turn. We will refer to  $A$  as the underlying matrix of  $U$  and to  $M$  as its turn. We will use  $U, V$  to range over U-matrices.

It turns out that there is a very natural algebra of U-matrices that extends that of ordinary matrices. We develop this algebra below.

Given an  $m \times n$  U-matrix  $V = (B, [L])$  and  $n \times p$  U-matrix  $U = (A, [K])$  their multiplication, written  $VU$ , is defined as the  $m \times p$  U-matrix:

$$(BA, [K + A^T LA])$$

**Lemma 3.4** Multiplication of U-matrices is well-defined.

**Proof.** Suppose that  $K \bowtie K'$  and  $L \bowtie L'$ . We need  $K + A^T LA \bowtie K' + A^T L'A$ .

$$\begin{aligned} K + A^T LA + (K + A^T LA)^T &= K + K^T + A^T LA + A^T L^T A \\ &= K + K^T + A^T(L + L^T)A \\ &= K' + K'^T + A^T(L' + L'^T)A \\ &= K' + A^T L'A + (K' + A^T L'A)^T \end{aligned}$$

□

**Lemma 3.5** Multiplication of U-matrices is associative.

**Proof.** This is true of the underlying matrices, hence it suffices to check associativity on the turns. Suppose that  $W = (C, [M])$ ,  $V = (B, [L])$ ,  $U = (A, [K])$  are U-matrices on which  $WV$  and  $VU$  are defined. Then their turns are, respectively,  $[L + B^T MB]$  and  $[K + A^T LA]$ . The turn of  $(WV)U$  is thus the equivalence class of

$$\begin{aligned} K + A^T(L + B^T MB)A &= K + A^T LA + A^T B^T MBA \\ &= K + A^T LA + (BA)^T M(BA) \end{aligned}$$

which is matrix that induces the turn of  $W(VU)$ . □

The  $n \times n$  identity U-matrix is  $(I_n, [0_n])$  where  $I_n$  is an identity matrix and  $0_n$  is a zero matrix. We abuse notation by also referring to the identity U-matrix as  $I_n$ .

**Lemma 3.6** Suppose that  $U$  is an  $m \times n$  U-matrix. Then  $I_m U = U I_n = U$ .

**Proof.** Suppose  $U = (A, [K])$ . Clearly it suffices to focus on the turns. The turn of  $I_m U$   $[K + A^T 0_m A] = [K]$ . The turn of  $U I_n$  is  $[0_n + I_n^T K I_n] = [K]$ . □

**Definition 3.7 (Category of U-matrices, UMat)** UMat has:

- as its set of objects, the set of natural numbers,
- and as arrows from  $n$  to  $m$ ,  $m \times n$  U-matrices.

Composition is U-matrix multiplication:  $U \circ V \stackrel{\text{def}}{=} UV$ , with identity arrows the identity U-matrices.

Given an  $m_1 \times n_1$   $\cup$ -matrix  $U = (A, [K])$  and  $m_2 \times n_2$   $\cup$ -matrix  $V = (B, [L])$ , their *direct sum*, written  $U \oplus V$ , is the  $(m_1 + m_2) \times (n_1 + n_2)$   $\cup$ -matrix

$$(A \oplus B, [K \oplus L]).$$

This is well-defined—if  $K \bowtie K'$  and  $L \bowtie L'$  then clearly  $K \oplus L \bowtie K' \oplus L'$ . The direct sum yields a strict monoidal product on  $\mathbf{UMat}$ .

There is an faithful identity-on-objects monoidal functor  $U : \mathbf{Mat} \rightarrow \mathbf{UMat}$  defined  $U(A) = (A, [0_n])$ . We have seen that  $\mathbf{Mat}$  is a PROP (Definition 2.1); in fact,  $\mathbf{UMat}$  is also a PROP, inheriting its permutations from  $\mathbf{Mat}$  via  $U$ .

**Lemma 3.8**  *$\mathbf{UMat}$  is a PROP.*

**Proof.** It suffices to show that symmetries are natural, i.e. the following diagram commutes for any  $m' \times m$   $\cup$ -matrix  $U = (A, [K])$  and  $n' \times n$   $\cup$ -matrix  $V = (B, [L])$ .

$$\begin{array}{ccc} m + n & \xrightarrow{\sigma_{m,n}} & n + m \\ U \oplus V \downarrow & & \downarrow V \oplus U \\ m' + n' & \xrightarrow{\sigma_{m',n'}} & n' + m' \end{array}$$

On the underlying matrices we clearly have  $\sigma_{m',n'}(A \oplus B) = (B \oplus A)\sigma_{m,n}$ . Notice that  $\sigma_{k,l}^T = \sigma_{l,k}$  for any  $k, l \in \mathbb{N}$ . We can use this to show that the turns agree:

$$\sigma_{m,n}^T(L \oplus K)\sigma_{m,n} = \sigma_{n,m}(L \oplus K)\sigma_{m,n} = K \oplus L$$

□

Indeed, the functor  $U : \mathbf{Mat} \rightarrow \mathbf{UMat}$  is a PROP homomorphism.

We end this section with a simple, yet useful way to factorise  $\cup$ -matrices. By a *pure turn* we mean an arrow of type  $m \rightarrow 0$  in  $\mathbf{UMat}$ , i.e. a  $0 \times m$   $\cup$ -matrix. By a *turn-free*  $\cup$ -matrix we mean an arrow in the image of  $U : \mathbf{Mat} \rightarrow \mathbf{UMat}$ , i.e., a  $\cup$ -matrix of the form  $(A, [0])$ . In the following, let  $\Delta_n$  denote the turn-free  $(n + n) \times n$   $\cup$ -matrix  $\left(\begin{pmatrix} I_n \\ I_n \end{pmatrix}, [0_n]\right)$ .

**Lemma 3.9** *Suppose that  $U$  is an  $m \times n$   $\cup$ -matrix. Then*

$$U = (A \oplus P)\Delta_n$$

where  $P$  is a pure turn and  $A$  is turn free.

**Proof.** Easy calculation:  $(A, [K]) = ((A, [0]) \oplus (!, [K]))\Delta_n$ . □

## 4 $\cup$ -Matrices as a Symmetric Monoidal Theory

The goal of this section is to give a presentation of  $\mathbf{UMat}$ , extending the presentation of  $\mathbf{Mat}$  recalled in §2. We introduce a symmetric monoidal theory,  $\mathbf{HAU}$ , and prove that  $\mathbf{HAU} \cong \mathbf{UMat}$  as PROPs.

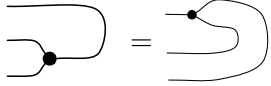
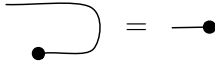

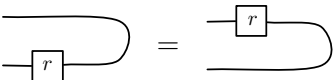
	$\cup \circ (id \oplus \nabla) = \cup \circ (id \oplus \cup \oplus id) \circ (\Delta \oplus id_2) \quad (\text{B1})$
	$\cup \circ (id \oplus \top) = \perp \quad (\text{B2})$
	$\cup \circ tw = \cup \quad (\text{B3})$
	$\cup \circ (id \oplus r) = \cup \circ (r \oplus id) \quad (\text{B4})$

Fig. 2. Additional equations of HAU.

**Definition 4.1 (HAU)** *The symmetric monoidal theory HAU is the free PROP on generators  $\{\Delta : 1 \rightarrow 2, \perp : 1 \rightarrow 0, \nabla : 2 \rightarrow 1, \top : 0 \rightarrow 1, \cup : 2 \rightarrow 0\} \cup \{r : 1 \rightarrow 1\}_{r \in R}$  and the set of equations listed in Fig. 1 together with the additional equations concerning  $\cup$  (pronounced “cup”) listed in Fig. 2. In string diagrams,  $\cup$  is drawn*

$$\cup.$$

Our main goal for this section is to prove the following.

**Theorem 4.2** *Suppose that  $R$  satisfies the condition  $2r = 2s$  implies  $r = s$ , for all  $r, s \in R$ . Then  $\text{HAU} \cong \mathbf{UMat}$ .*

The remainder of this section consists of a proof of the above result; we thus assume that  $R$  satisfies the required condition for the remainder of this section.

We start by defining a PROP morphism  $\Theta : \text{HAU} \rightarrow \mathbf{UMat}$  recursively.

$$\begin{aligned} \Delta &\mapsto \left( \begin{pmatrix} 1 \\ 1 \end{pmatrix}, [(0)] \right) & \nabla &\mapsto \left( \begin{pmatrix} 1 & 1 \end{pmatrix}, \left[ \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \right] \right) & \cup &\mapsto \left( !, \left[ \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \right] \right) \\ \perp &\mapsto (!, [(0)]) & \top &\mapsto (i, [()]) \end{aligned}$$

Note that we are being somewhat sloppy with notation since we denote by  $!$  all unique arrows to 0 and by  $i$  all unique arrows from 0 in  $\mathbf{Mat}$ . Since we are defining a symmetric monoidal functor, the recursion is forced, e.g.:

$$\Theta(q \circ p) = \Theta(q) \circ \Theta(p) \quad \Theta(p \oplus q) = \Theta(p) \oplus \Theta(q)$$

This procedure is well-defined: since both  $\text{HAU}$  and  $\mathbf{UMat}$  are symmetric monoidal it suffices to show that all equations in  $\text{HAU}$  hold also in  $\mathbf{UMat}$ . This is the case for  $\text{HA}$  and  $\mathbf{Mat}$  (as shown in [3]) – it suffices to check the equations involving  $\cup$ .

**Lemma 4.3**

- (i)  $(\Theta \cup)(\Theta \nabla \oplus id) = (\Theta \cup)(id \oplus \Theta \cup \oplus id)(id_2 \oplus \Theta \Delta)$
- (ii)  $(\Theta \cup)(id \oplus \Theta \top) = \Theta \perp$
- (iii)  $(\Theta \cup)(\Theta tw) = \Theta \cup$
- (iv)  $(\Theta \cup)\Theta(id \oplus r) = (\Theta \cup)\Theta(r \oplus id)$ .



**Proof.** Straightforward calculations.  $\square$

**Lemma 4.4** *The following diagram commutes.*

$$\begin{array}{ccc}
 \mathbf{HA} & \longrightarrow & \mathbf{HAU} \\
 \Phi \downarrow & & \downarrow \Theta \\
 \mathbf{Mat} & \xrightarrow{U} & \mathbf{UMat}
 \end{array} \tag{1}$$

**Proof.** Since the diagram consists of PROP homomorphisms, it suffices to verify that it is the case for the generators of  $\mathbf{HA}$ .  $\square$

The above observation allows us to conclude that the additional axioms of  $\mathbf{HAU}$  are conservative wrt the equations in  $\mathbf{HA}$ .

**Lemma 4.5** *Suppose  $t, u \in \mathbf{HA}$ . Then  $t =_{\mathbf{HA}} u$  iff  $t =_{\mathbf{HAU}} u$ .*

**Proof.** The ‘only if’ direction is obvious. For the ‘if’, observe that in the diagram (1),  $\Phi$  is an isomorphism and  $U$  is faithful, so their composition is faithful. Since the diagram commutes, the homomorphism  $\mathbf{HA} \rightarrow \mathbf{HAU}$  is faithful.  $\square$

Since  $\Theta$  is identity on objects, to show it is an isomorphism it suffices to show that it is full and faithful.

**Proposition 4.6**  *$\Theta$  is full.*

**Proof.** Suppose that  $U = (A, [K])$  is an  $m \times n$   $\cup$ -matrix. We must construct a term  $t_U$  of  $\mathbf{HAU}$  such that  $\Theta(t_U) = U$ . Using the factorisation of Lemma 3.9, we have  $U = ((A, [0_n]) \oplus (!, [K]))\Delta_n$ . We can break up the problem into two parts: construct  $t_A : n \rightarrow m$  such that  $\Theta(t_A) = (A, [0_n])$  and  $t_K : n \rightarrow n$  such that  $\Theta(t_K) = (\text{id}_n, [K])$ . Since  $\mathbf{HA} \cong \mathbf{Mat}$ , it is straightforward to obtain  $t_A$ .

We are left with  $(!, [K])$ , where w.l.o.g. we assume that  $K$  is upper triangular. The problem becomes: given an upper triangular  $n \times n$   $K$ , construct a term of  $t_K$  such that  $\Theta(t_K) = (\text{id}_n, [K])$ . We can do this by constructing two terms in  $\mathbf{HAU}$ :

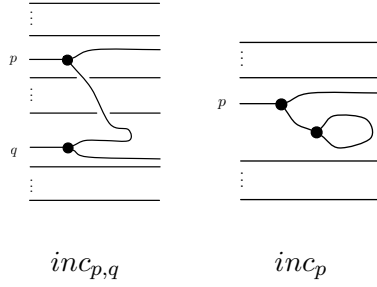
- given  $p < q \leq n$ , a term  $\text{inc}_{p,q} : n \rightarrow n$ , satisfying, for any  $(B, [L])$

$$\Theta(\text{inc}_{p,q})(B, [L]) = (B, [L']) \text{ where } L'_{ij} = \begin{cases} L_{ij} + 1 & \text{if } i = p \text{ and } j = q \\ L_{ij} & \text{otherwise.} \end{cases}$$

- given  $p \leq n$ ,  $\text{inc}_p : n \rightarrow n$ , satisfying,

$$\Theta(\text{inc}_p)(B, [L]) = (B, [L']) \text{ where } L'_{ij} = \begin{cases} L_{ij} + 1 & \text{if } i = j = p \\ L_{ij} & \text{otherwise.} \end{cases}$$

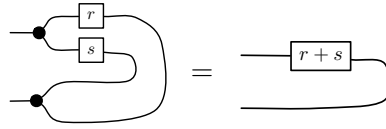
For sake of simplicity, we only give the appropriate string diagrams below and let the reader verify that  $\Theta(\text{inc}_{p,q})$  and  $\Theta(\text{inc}_p)$  work as advertised:



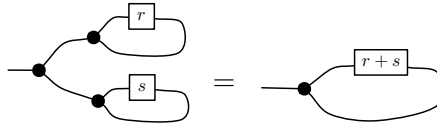
Then, clearly,  $t_K$  can be constructed by an appropriate composition of these two families of terms, finally composing with  $\perp_n$  to obtain  $t_K$ .  $\square$

Proving that  $\Theta$  is faithful is slightly more involved. We start by showing how turns “add” in HAU; the proofs are simple exercises in diagrammatic reasoning.

**Lemma 4.7**



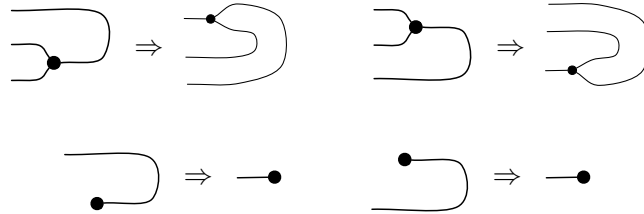
**Lemma 4.8**



The crux of the proof is a certain “normal form” for representations of pure turns in HAU. Basically, one can write any such term *without* using  $\nabla$  or  $\top$ .

**Lemma 4.9** *Suppose that  $t : k \rightarrow 0$  in HAU. Then  $t$  can be written  $t_{\cup} \circ t_{\Delta}$  where  $t_{\Delta}$  uses only generators  $\Delta, \perp$ , and  $t_{\cup}$  only  $\cup$ .*

**Proof.** We use a rewriting argument: start with a string diagrammatic representation of  $t$ . Next, we can write  $t = t_{\cup}^1 \circ t_{\text{HA}}^1$  where  $t_{\text{HA}}^1$  does not use the  $\cup$  generator and  $t_{\cup}^1$  only the  $\cup$  generator. Now  $t_{\text{HA}}^1$  can be put in *matrix form* (see [3]): a composition  $t_{\nabla}^1 \circ t_{\Delta}^1$  where all comonoid structure comes before the monoid structure. The following are the rewrite rules that we will apply:



at each point obtaining a step  $t_{\cup}^k \circ t_{\text{HA}}^k \Rightarrow t_{\cup}^{k+1} \circ t'_{\text{HA}} \Rightarrow t_{\text{HA}}^{k+1} \circ t_{\cup}^{k+1}$  where  $t_{\text{HA}}^{k+1}$  results from  $t'_{\text{HA}}$  by putting it in matrix form.

The only tricky issue is proving termination: we can show it using the concept of *monoid depth*. Given a term  $t_{\text{HA}} : k \rightarrow l$ , we let the monoid depth of right port  $0 \leq i < l$  of  $md(t_{\text{HA}}, i)$  be the number of non-zero entries in the  $i$ th row of the

matrix that corresponds to  $t_{\mathbf{HA}}$ . The isomorphism  $\mathbf{HA} \cong \mathbf{Mat}$  guarantees that this definition makes sense. The goal is to reach a term where all ports have monoid depth 1, i.e. one whose matrix has exactly one non-zero entry in each row.

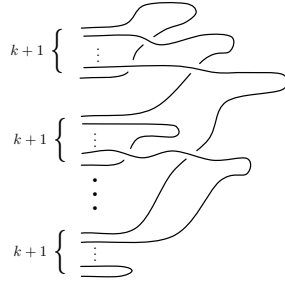
Now, given two ports with monoid depth  $k > 1, l$  connected by a turn, each of the first two rewrite rules decrements the monoid depth of the  $k$  port, and adds another port of depth  $l$ . The last two rules simply remove a port of multiplication depth 0 (in other words, remove a zero row from the matrix). By always rewriting at a port with the highest monoid depth, and cancelling any ports of depth 0, clearly one eventually arrives at a term with all ports of monoid depth 1.  $\square$

**Lemma 4.10 (Turn matrix form)** *Any  $t : k \rightarrow 0$  can be written in the form*

$$t = \Upsilon_k \circ \left( \bigoplus_{1 \leq i \leq k} \bigoplus_{1 \leq j \leq k+1} r_{i,j} \right) \circ \left( \bigoplus_{1 \leq i \leq k} \Delta_{k+1} \right)$$

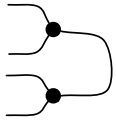
where  $r_{i,j} = 1$  whenever  $j = k + 1$  or  $i > j$  and  $\Upsilon : k(k + 1) \rightarrow 0$  is constructed by connecting, for each  $1 \leq i \leq k$ :

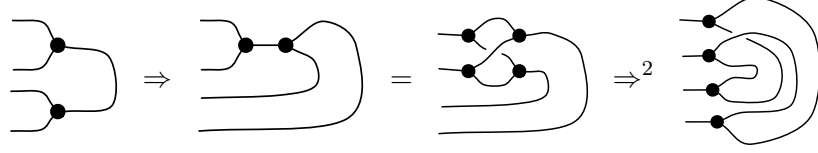
- the  $(i - 1)(k + 1) + i$ th port and  $((i - 1)(k + 1) + k + 1)$ th port, and
- for each  $j > i$ , the  $(i - 1)(k + 1) + j$ th port with the  $(j - 1)(k + 1) + i$ th port with cups. A recursive definition can be given, but it is unenlightening: the diagram below does a better job of illustrating the structure of  $\Upsilon_k$ .



Let  $K$  be the upper triangular matrix with  $j, i$ th entry  $r_{ij}$  when  $i \leq j$  and 0 otherwise. Then  $\Theta(t) = (1, [K])$ .

**Proof.** Using the procedure of Lemma 4.9 we can write any term  $k \rightarrow 0$  in the form  $t_{\cup} \circ t_{\Delta}$ , then add up the coefficients using the equations of Lemmas 4.8 and 4.7.  $\square$

**Example 4.11** The turn matrix form of  is calculated below:



where we did not draw the 0-coefficient turns. It is easily checked that this gives the upper triangular matrix

$$\begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

**Proposition 4.12**  $\Theta$  is faithful.

**Proof.** Suppose that  $t, u : n \rightarrow m$  such that  $\Theta(t) = \Theta(u)$ . Using the laws of HA we can write  $t = (t_{\text{HA}} \oplus t') \circ \Delta_n$  and  $u = (u_{\text{HA}} \oplus u') \circ \Delta_n$ , where  $t_{\text{HA}}, u_{\text{HA}} : n \rightarrow m$  and  $t', u' : n \rightarrow 0$ .

Now  $\Theta(\Delta_n)$  is mono, thus  $\Theta(t) = \Theta(u)$  implies

$$\Theta(t_{\text{HA}} \oplus t') = \Theta(u_{\text{HA}} \oplus u').$$

By definition of direct sum in  $\mathbf{UMat}$ , we have

$$\Theta(t_{\text{HA}}) = \Theta(u_{\text{HA}}) \quad \Theta(t') = \Theta(u')$$

Since  $\text{HA} \cong \mathbf{Mat}$  and  $U : \mathbf{Mat} \rightarrow \mathbf{UMat}$  is faithful, we have  $t_{\text{HA}} = u_{\text{HA}}$ . It remains to show that  $\Theta(t') = \Theta(u')$  implies that  $t' = u'$ . Transform  $t'$  and  $u'$  into turn matrix form. Since  $(!, [K]) = \Theta(t') = \Theta(u') = (!, [L])$ , we must have  $[K] = [L]$ . Since  $K$  and  $L$  are upper triangular, the conclusion of Lemma 3.2 yields  $K = L$ . Thus  $t' = u'$ .  $\square$

We have shown that  $\Theta$  is full in Proposition 4.6, and that it is faithful in Proposition 4.12. This completes the list of ingredients needed for the proof of Theorem 4.2.

## 5 A compositional theory of simple graphs: ABUV

In this section we focus on the case  $R = \mathbb{Z}_2$ . We give a presentation for  $\mathbf{UMat}_{\mathbb{Z}_2}$ , and show how one can use this to obtain an algebra of simple graphs.

### 5.1 $\cup$ -Matrices over $\mathbb{Z}_2$

**Definition 5.1 (AB and ABU)** The symmetric monoidal theory AB is the free PROP on generators  $\{\Delta : 1 \rightarrow 2, \perp : 1 \rightarrow 0, \nabla : 2 \rightarrow 1, \top : 0 \rightarrow 1\}$ , and equations (A1)-(A10) listed in Fig. 1 and (C1) in Fig. 3.

$$\begin{array}{l} \text{---} \bullet \begin{array}{c} \text{---} \text{---} \\ \text{---} \end{array} \bullet \text{---} = \text{---} \bullet \text{---} \quad \nabla \circ \Delta = \top \circ \perp \quad (\text{C1}) \\ \text{---} \bullet \begin{array}{c} \text{---} \\ \text{---} \end{array} \bullet = \text{---} \bullet \quad (\cup \circ \Delta = \perp) \quad (\text{C2}) \end{array}$$

Fig. 3. Equations particular to ABU and ABUV.

The symmetric monoidal theory ABU is the free PROP on generators  $\{\Delta : 1 \rightarrow 2, \perp : 1 \rightarrow 0, \nabla : 2 \rightarrow 1, \top : 0 \rightarrow 1, \cup : 2 \rightarrow 0\}$ , the equations of AB, equations (B1)-(B3) of Fig. 2 and (C2) in Fig. 3.

First recall [2, 15] that  $\text{AB} \cong \text{Mat}_{\mathbb{Z}_2}$ . Likewise, presenting  $\text{UMat}_{\mathbb{Z}_2}$  is simpler than the general case: there is no need for additional scalars, and the equation  $1 + 1 = 0$  is captured by the “anti-separable” law – equation (C1) in Fig. 3 [2]. The equation (C2) may seem surprising, since it does not hold in the general case. Indeed, the requirement of “no self loops” in simple graphs is already taken care of by the algebra of  $\cup$ -matrices with  $\mathbb{Z}_2$  entries.

**Lemma 5.2**  $\Theta(\cup \circ \Delta) = \Theta(\top)$

**Proof.** For  $\mathbb{Z}_2$ -matrices we have  $(0) \bowtie (1)$ , so  $[(0)] = [(1)]$ .  $\square$

**Theorem 5.3**  $\text{ABU} \cong \text{UMat}_{\mathbb{Z}_2}$ .

**Proof.** Proceeds along the lines of the proof of Theorem 4.2: the only difference is that instead of using upper triangular matrices as representatives of turns, we use upper triangular matrices with 0 diagonals. We omit the details.  $\square$

## 5.2 An algebra of simple graphs

We will use the theory  $\text{ABU}(\cong \text{UMat}_{\mathbb{Z}_2})$  as an adjacency algebra of simple graphs. The only thing missing is the vertices.

**Definition 5.4 (ABUV)** The symmetric monoidal theory ABUV is the free PROP on generators  $\{\Delta : 1 \rightarrow 2, \perp : 1 \rightarrow 0, \nabla : 2 \rightarrow 1, \top : 0 \rightarrow 1, \cup : 2 \rightarrow 0, v : 0 \rightarrow 1\}$ , and the equations of ABU. In string diagrams, we will draw  $v : 0 \rightarrow 1$  as follows:



Since there are no additional equations involving  $v$ , ABUV is the coproduct (in the category of PROPs) of ABU and the free PROP on generator  $v$  and no equations.

**Definition 5.5** Let  $\text{CGraph}$  be the 2-PROP with arrows  $n \rightarrow m$  pairs  $(k, U)$  where

- $k \in \mathbb{N}$  and
- $U$  is an  $m \times (n + k)$   $\cup$ -matrix.

Composition  $(V, l) \circ (U, k)$  is

$$(k + l, V(U_1 \oplus I_k))$$

For the 2-cells, given a bijection  $\sigma : k \rightarrow k$ , let  $I_\sigma$  denote the permutation matrix

induced by  $\sigma$ . Then  $\sigma : (U, k) \Rightarrow (U', k) : n \rightarrow m$  is a 2-cell if

$$U'(I_n \oplus I_\sigma) = U.$$

Note that all 2-cells are invertible. Let  $\mathbf{CGraph}_{\cong}$  denote the category obtained by identifying isomorphic 1-cells in  $\mathbf{CGraph}$ .

**Theorem 5.6**  $\text{ABUV} \cong \mathbf{CGraph}_{\cong}$ .

**Proof.** Omitted. □

**Corollary 5.7** There is a 1-1 correspondence between  $\text{ABUV}[0, 0]$  and (isomorphism classes of) simple graphs.

**Proof.** A simple graph  $G$  on  $k$  vertices can be identified with an upper-triangular  $k \times k$   $\mathbb{Z}_2$  matrix  $M_G$  with 0s on the diagonal—its adjacency matrix. Two simple graphs  $G, G'$  are isomorphic precisely when there exists bijection  $\sigma : k \rightarrow k'$  s.t.  $M_{G'} = I_\sigma^T M_G I_\sigma$ . The statement thus follows directly from Theorem 5.6. □

We conclude the paper with an illustrative, yet simple example of how the algebra of  $\text{ABUV}$  can be used to decompose graphs. It is well-known that cliques have unbounded tree-width, but are very simple from the point of view of rank-width: their rank-width is 1. This is easy to observe: suppose that  $\{V_1, V_2\}$  is any partition of the vertex set  $V$  of a clique. Then the  $|V_2| \times |V_1|$  adjacency matrix that describes how vertices of  $V_1$  connect to the vertices of  $V_2$  is simply the matrix with 1 in each entry, and this matrix has rank 1.

Cliques also enjoy a very simple description in  $\text{ABUV}$ , as we show in the following.

**Example 5.8** [Clique] The clique on  $n$  vertices can be constructed as follows

$$\perp \circ c^n \circ \top$$

where  $c$  is the following term:



Here we do not define “simplicity”, which is measured with a structural metric called *monoidal width*; it will be introduced in a sequel to this paper. The basic idea is the following: a *monoidal decomposition* is simply a binary tree with internal nodes labelled with  $\circ$  or  $\oplus$ , and leaves generators, identities and twists. The *width* associated to the decomposition tied to the *largest object* along which a composition is ever performed when evaluating the tree as a term of the ambient symmetric monoidal theory. Monoidal width of a term is then the width of an optimal decomposition, i.e. one that has the smallest width.

As we shall show in the sequel, monoidal width in  $\text{ABUV}$  is closely related to rank-width, and concepts such as tree-width can be characterised in a similarly natural way by restricting to sub-PROPs of  $\text{ABUV}$ . Note also that the definition of

monoidal width is not tied to ABUV and makes sense in any monoidal theory, which solves the problem of generality outlined in the Introduction. Our claims are that (i) ABUV is a canonical, compositional algebra of simple graphs, and (ii) monoidal width is a robust concept that we believe will be useful in a number of different applications areas [1, 4, 7, 10, 19] where monoidal theories are used.

## References

- [1] F. Bonchi, P. Sobociński, and F. Zanasi. A categorical semantics of signal flow graphs. In *CONCUR'14*, 2014.
- [2] F. Bonchi, P. Sobociński, and F. Zanasi. Interacting bialgebras are Frobenius. In *FoSSaCS '14*, volume 8412 of *LNCS*, pages 351–365. Springer, 2014.
- [3] F. Bonchi, P. Sobociński, and F. Zanasi. Interacting Hopf algebras. *submitted, available at <http://arxiv.org/abs/1403.7048>*, 2014.
- [4] F. Bonchi, P. Sobociński, and F. Zanasi. Full abstraction for signal flow graphs. In *Principles of Programming Languages, POPL'15.*, 2015.
- [5] B. Coecke and R. Duncan. Interacting quantum observables. In *ICALP'08*, pages 298–310, 2008.
- [6] B. Courcelle and J. Engelfriet. *Graph Structure and Monadic Second-Order Logic - A Language-Theoretic Approach*. Number 138 in *Encyclopedia of mathematics and its applications*. CUP, 2012.
- [7] R. Duncan. A graphical approach to measurement-based quantum computing. In *Quantum Physics and Linguistics: A Compositional, Diagrammatic Discourse*, chapter 3. OUP, 2013.
- [8] M. P. Fiore and M. D. Campos. The algebra of directed acyclic graphs. In *Computation, Logic, Games, and Quantum Foundations. The Many Facets of Samson Abramsky*, volume 7860 of *LNCS*, 2013.
- [9] F. Gadducci and R. Heckel. An inductive view of graph transformation. In *Recent Trends in Algebraic Development Techniques*, volume 1376 of *LNCS*. Springer, 1998.
- [10] D. R. Ghica. Diagrammatic reasoning for delay-insensitive asynchronous circuits. In *Computation, Logic, Games, and Quantum Foundations. The Many Facets of Samson Abramsky*, pages 52–68. Springer, 2013.
- [11] P. Katis, N. Sabadini, and R. F. C. Walters. Bicategories of processes. *J. Pure Appl. Algebra*, 115:141–178, 1997.
- [12] P. Katis, N. Sabadini, and R. F. C. Walters. Span(Graph): an algebra of transition systems. In *Algebraic Methodology and Software Technology (AMAST '97)*, volume 1349 of *LNCS*, pages 322–336. Springer, 1997.
- [13] S. La Torre and G. Parlato. Scope-bounded multistack pushdown systems: Fixed-point, sequentialization, and tree-width. In *FSTTCS 2012*, 2012.
- [14] S. Lack. Composing PROPs. *Theor App Categories*, 13(9):147–163, 2004.
- [15] Y. Lafont. Towards an algebraic theory of boolean circuits. *J Pure Appl Alg*, 184:257–310, 2003.
- [16] S. Mac Lane. Categorical algebra. *Bull Amer Math Soc*, 71:40–106, 1965.
- [17] S.-I. Oum. *Graphs of bounded rank-width*. PhD thesis, Princeton, 2005.
- [18] S.-I. Oum and P. Seymour. Approximating clique-width and branch-width. *Journal of Combinatorial Theory, Series B*, 96(4):514 – 528, 2006.
- [19] J. Rathke, P. Sobociński, and O. Stephens. Compositional reachability in petri nets. In *Reachability Problems*, volume 8762 of *LNCS*, pages 230–243. Springer, 2014.
- [20] P. Sobociński. Representations of Petri net interactions. In *Concurrency Theory (CONCUR '10)*, number 6269 in *LNCS*, pages 554–568. Springer, 2010.
- [21] P. Sobociński. Nets, relations and linking diagrams. In *Algebra and Coalgebra in Computer Science (CALCO '13)*, volume 8089 of *LNCS*, pages 282–298, 2013.