Compositional Modelling of Network Games

Elena Di Lavore
Department of Software Science, Tallinn University of Technology, Estonia
elena.di@taltech.ee

Jules Hedges
Department of Computer and Information Sciences, University of Strathclyde, Glasgow, UK
jules.hedges@strath.ac.uk

Paweł Sobociński
Department of Software Science, Tallinn University of Technology, Estonia
pawel.sobocinski@taltech.ee

Abstract

The analysis of games played on graph-like structures is of increasing importance due to the prevalence of social networks, both virtual and physical, in our daily life. As well as being relevant in computer science, mathematical analysis and computer simulations of such distributed games are vital methodologies in economics, politics and epidemiology, amongst other fields. Our contribution is to give compositional semantics of a family of such games as a well-behaved mapping, a strict monoidal functor, from a category of open graphs (syntax) to a category of open games (semantics). As well as introducing the theoretical framework, we identify some applications of compositionality.

2012 ACM Subject Classification Theory of computation → Categorical semantics

Keywords and phrases game theory, category theory, network games, open games, open graphs, compositionality

Digital Object Identifier 10.4230/LIPIcs.CSL.2021.30

Funding *Elena Di Lavore* and *Paweł Sobociński* are supported by the ESF funded Estonian IT Academy research measure (project 2014-2020.4.05.19-0001).

1 Introduction

Compositionality concerns finding homomorphic mappings

$$\textbf{Syntax} \to \textbf{Semantics}. \tag{1}$$

This important concept originated in formal logic [20, 21], and is at the centre of formal semantics of programming languages [23]. In recent years, there have been several *2-dimensional* examples [6, 4, 1], where both **Syntax** and **Semantics** are symmetric monoidal categories. Usually **Syntax** is freely generated from a (monoidal) signature, possibly modulo equations. This opens up the possibility of recursive definitions and proofs by structural induction, familiar from our experience with ordinary, 1-dimensional syntax.

In this paper, we consider an instance of (1) that is—at first sight—quite different from the usual concerns of programming and logic: network games [5], also known as graphical games. Network games involve agents that play concurrently, and share information based on an underlying, ambient network topology. Indeed, the utility of each player typically depends on the structure of the network. An interesting application is social networks [10], but they also feature in economics [11], politics [22] and epidemiology [16], amongst other fields. (These games should not be confused with classes of dynamic games played on graphs, such as parity games and pursuit games, which are not within the scope of this paper.)

© Elena Di Lavore, Jules Hedges, and Paweł Sobociński;
licensed under Creative Commons License CC-BY

In formal accounts of network games, graphs represent network topologies. Players are identified with graph vertices, and their utility is influenced only by their choices and those of their immediate neighbours. Network games are thus “games on graphs”. An example is the *majority game*: players “win” when they agree with the majority of their neighbours.

In what way do such games fit into the conceptual framework of (1)? Our main contribution is the framing of certain network games as monoidal functors from a suitable category of *open graphs* **Grph** [7], our **Syntax**, to the category of open games **Game** [13], our **Semantics**. Given a network game $\mathcal{N}$ (e.g. the majority game), such games are functors

$$F_{\mathcal{N}} \colon \mathbf{Grph} \to \mathbf{Game} \tag{2}$$

that, for any *closed* graph $\Gamma \in \mathbf{Grph}$, yield the game $F_{\mathcal{N}}(\Gamma)$, which is the game $\mathcal{N}$ played on Γ. However, compositionality means that such games are actually “glued together” from simpler, *open* games. In fact, $F_{\mathcal{N}}$ maps each vertex of Γ to an open game called the *utility-maximising player*, and the connectivity of Γ is mapped, following the rules of $\mathcal{N}$, to structure in **Game**.

Our contribution thus makes the intuitively obvious idea that the data of network games is dependent on their network topology precise. Concrete descriptions of network games, given a fixed topology, are often quite involved: our approach means that they can be derived in a principled way from basic building blocks. In some cases, the compositional description can also help in the mathematical analysis of games. For example, in the case of the majority game, the right decomposition of a network topology Γ as an expression in **Grph** can yield a recipe for the Nash equilibrium of $F_{\mathcal{N}}(\Gamma)$ in **Game** in terms of the equilibria of the open games obtained via $F_{\mathcal{N}}$ from the open graphs in the decomposition. As it happens when solving optimization problems, a compositional analysis of the equilibria is possible only when the game has optimal substructure, which is the case for the majority game (but is not the case in general). Nevertheless, compositional modelling is valuable for the understanding of the structure of the system. It allows, for example, to modify a part of a system while keeping the analysis done for the rest of the system, as we show in Example 31.

Technically, we proceed as follows. We introduce *monoid network games* (Definition 7) that make common structure of all of our motivating examples explicit, and that we believe cover the majority of network games studied in the literature. Roughly speaking, monoid network games are parametrised wrt *(i)* a monoid that aggregates information from neighbours and *(ii)* functions that govern how that information is propagated in the network. While we are able to model all network games, the structure of monoid network games allows us to characterise them as functors in a generic fashion.

Our category of open graphs **Grph** (Definition 18) is an extension of the approach of [7], from undirected graphs to undirected *multi*graphs. Multigraphs allow us to model games on networks where some links are stronger than others, cf. Example 30. Our **Grph** is different from other notions of “open graph” in the literature, e.g. via cospans [9], in that it is centred on the use of *adjacency matrices*, which are commonly used in graph theory to encode connectivity. Adjacency matrices give an explicit presentation of the graphs that allows an explicit description of the games played on them. Moreover, the emphasis on the matrix algebra means that **Grph** has the structure of commutative bialgebra—equivalent to the algebra of ordinary $\mathbb{N}$ matrices [15, 24]—but also additional structure that captures the algebraic content of adjacency matrices. Given that **Grph** has a presentation in terms of generators and equations, to obtain (2) it suffices to define it on the generators and check that **Grph**-equations are respected in **Game**. This is our main result, Theorem 27.

In addition to the presentation of **Grph** in terms of generators and equations, we characterise it as another category (Theorem 23) that makes clear its status as a category of "open graphs". The result can be understood as a kind of normal form for the morphisms of **Grph**, useful to describe concrete instantiations of $F_{\mathcal{N}}$ for arbitrary open graphs (Theorem 29).

Our work is a first step towards a more principled way of defining games parametrised by graphs. We would like to remark that the methodology that we present to define games on networks is more general than the particular instance worked out in this paper. Indeed, future work will extend both the notions of graphs (e.g. by considering directed graphs), as well as the kinds of games played on them (e.g. stochastic games, repeated games). While we do identify some applications, we believe that compositional reasoning is severely under-rated in traditional game theory, and that its adoption will lead to both more flexible modelling frameworks, as well as more scalable mathematical analyses.

Structure of the paper

We introduce our running examples in §2 and unify them under the umbrella of monoid network games. Next, we recall the basics of open games in §3 and identify the building blocks needed for (2). In §4 we introduce the category **Grph** of open, undirected multigraphs, and give a combinatorial characterisation, which is useful in applications. The construction of $F_{\mathcal{N}}$ is in §5, and several applications of our compositional framework are given in §6.

2 Network games

In this section we introduce motivating examples for our compositional framework and introduce a notion of game called the *monoid network game* that unifies them.

Network games [5, 14] are parametric wrt a network topology, usually represented by a graph. Players are the vertices, and the possible connections between the players are represented by the edges. Moreover, each player's payoff is affected only by the choices of its immediate *neighbours* on the graph. We use undirected multigraphs to model network topologies.

▶ **Definition 1.** *An undirected multigraph is $G = (V_G, E_G)$, where V_G is the set of vertices and E_G is a sym. multi-relation on V_G: a function $E_G \colon V_G \times V_G \to \mathbb{N}$ st $E_G(v_i, v_j) = E_G(v_j, v_i)$.*

A common way of capturing the connectivity of a graph is via *adjacency matrices*, which play an important role in graph theory. They are also crucial for our compositional account.

Assuming an ordering on the set of vertices of a graph, square matrices A with entries from $\mathbb{N}$ can record connections between vertex i and j in A_{ij}: a 0-entry signifies no edge, and non-zero entries count the connections. Ordinary matrices are too concrete to uniquely represent connectivity since edges between i and j can be recorded in the (i, j)th entry or the (j, i)th entry. One could use symmetric matrices or triangular matrices. For us, it is better to equate matrices that encode the same connectivity: $A \sim A'$ iff $A + A^T = A' + A'^T$.

▶ **Definition 2.** *An* adjacency matrix *is an equivalence class $[A]$ of matrices with entries in the natural numbers. The equivalence relation is given by*

$$A \sim A' \iff A + A^T = A' + A'^T.$$

A finite multigraph can also be defined as $(k_G, [A])$ where $k_G \in \mathbb{N}$ and $[A]$ a $k_G \times k_G$ adjacency matrix. Let $\mathbf{G}(n)$ be the set of multigraphs with n vertices, enumerated as $v_1, \ldots, v_n$.

▶ **Definition 3** (Network game). *An n-player* network game $\mathcal{N}$ *consists of, for each player $1 \le i \le n$, a set of choices X_i and a payoff $u_i : \mathbf{G}(n) \times \prod_{j=1}^{n} X_j \to \mathbb{R}$, such that each player's payoff is affected only by its own and its neighbours' choices: for each $G \in \mathbf{G}(n)$, each player i, each $j \neq i$ such that $(v_i, v_j) \notin E_G$, each $\underline{x}_{-j} \in \prod_{k \neq j}^{n} X_k$, and each $x_j, x'_j \in X_j$*

$$u_i(G, x_j, \underline{x}_{-j}) = u_i(G, x'_j, \underline{x}_{-j})$$

(The notation x_{-j}, standard in game theory, means a tuple with the jth element missing.)

The set of strategies *is $\prod_{i=1}^{n} X_i$ and its elements $\underline{x} \in \prod_{i=1}^{n} X_i$ are* strategy profiles.

The best response*, for a graph $G \in \mathbf{G}(n)$, is a relation $\mathbb{B}_{\mathcal{N}}$ on the set of strategies, defined by*

$$(\underline{x}, \underline{x}') \in \mathbb{B}_{\mathcal{N}} \Leftrightarrow \forall 1 \le i \le n.\ \forall y_i \in X_i.\ u_i(G, \underline{x}[i \mapsto x'_i]) \ge u_i(G, \underline{x}[i \mapsto y_i])$$

A Nash equilibrium*, for $G \in \mathbf{G}(n)$, is a strategy profile $\underline{x}$ s.t. for each player $1 \le i \le n$, $u_i(G, \underline{x}) \ge u_i(G, \underline{x}[i \mapsto x'_i])$ for each $x'_i \in X_i$. It is a fix-point of the best response relation.*

We now recall three important examples of network games.

▶ **Example 4** (Majority game). Each player has two choices, $X_i = \{Y, N\}$. A player receives a utility of 1 if its choice is the majority choice of its neighbours, and 0 otherwise, i.e.

$$u_i(G, \underline{x}) = \begin{cases} 1 & \text{if } |\{v_j \mid (v_i, v_j) \in E_G \text{ and } x_i = x_j\}| \ge |\{v_j \mid (v_i, v_j) \in E_G \text{ and } x_i \neq x_j\}| \\ 0 & \text{otherwise.} \end{cases}$$

Nash equilibria are strategy profiles where players take the majority choice of their neighbours.

▶ **Example 5** (Best-shot public goods game). Each player has two choices, $X_i = \{Y, N\}$, interpreted as investing or not investing in a public good. The investor bears a cost $0 < c < 1$, and gives a utility of 1 to themselves and every neighbour. The players are already partially satisfied with the current situation and assign a utility of $1 - c + \epsilon$, with $0 < \epsilon < c$, to the situation where neither the player nor its neighbours invest. The utility functions thus are:

$$u_i(G, \underline{x}) = \begin{cases} 1 - c & \text{if } x_i = Y \\ 1 & \text{if } x_i = N \text{ and } x_j = Y \text{ for some } (v_i, v_j) \in E_G \\ 1 - c + \epsilon & \text{otherwise.} \end{cases}$$

The Nash equilibrium is when no player invests, an example of a 'tragedy of the commons'.

▶ **Example 6** (Weakest-link public goods game). Each player's choice is an investment, valued in $\mathbb{R}_+$. The cost to the player given by an increasing cost function $c : \mathbb{R}_+ \to \mathbb{R}_+$ where $c(0) = 0$, and utility is the *minimum* level of investment of the player and all neighbours:

$$u_i(G, \underline{x}) = \min_{j = i \text{ or } (v_i, v_j) \in E_G} x_j - c(x_i).$$

A necessary condition for Nash equilibrium is that no player invests more than its neighbours.

In Examples 4, 5 and 6 every player has the same set of choices, and the utility depends in a uniform way on neighbours' choices. We collect these, and other examples in the literature, under the umbrella of *monoid network games.* Most examples in the literature can be collected in two classes [5, ch. 5], namely games on networks with constrained continuous actions or with binary actions. Provided that weights are natural numbers, the latter can be expressed as monoid network games. To express the former as monoid network games, we need to additionally ask that the parameters appearing in the utility functions of the players be constant. However, we can still express games of this class with different parameters for different players by composing different monoid network games. This is shown in Example 32.

Network games that do not fall into this category can be nevertheless expressed in a compositional way as illustrated in Fig. 1. If a game can be described in the form of a a *monoid* network game, we can say more: such games are a monoidal functor from the category of syntax to the category of semantics. The details are in Section 5.

To the best of our knowledge, the following has not previously appeared in the literature.

▶ **Definition 7** (Monoid network game). *A* monoid network game *is $\mathcal{N} = (X, M, f, g)$ where:*

- *X is the set of choices for each player*
- *$M = (M, \oplus, e)$ is a commutative monoid*
- *$f : X \to M$ and $g : X \times M \to \mathbb{R}$ are functions such that each utility function has the form*

$$u_i(G, \underline{x}) = g\left(x_i, \bigoplus_{(v_i,v_j)\in E_G} f(x_j)\right).$$

Examples 4, 5, 6 are indeed examples of monoid network games:

- The majority game (Example 4) has the monoid $(\mathbb{N}, +, 0) \times (\mathbb{N}, +, 0)$, counting the Y and N 'votes'. Define $f : \{Y, N\} \to \mathbb{N}^2$ by $f(Y) = (1, 0)$ and $f(N) = (0, 1)$, and $g : \{Y, N\} \times \mathbb{N}^2 \to \mathbb{R}$ is:

$$g(x, (n_1, n_2)) = \begin{cases} 1 & \text{if } x = Y \text{ and } n_1 \geq n_2 \\ 1 & \text{if } x = N \text{ and } n_1 \leq n_2 \\ 0 & \text{otherwise.} \end{cases}$$

- The best-shot public goods game (Example 5) is a monoid network game with the monoid $\textbf{Bool} = (\{Y, N\}, \vee, N)$, where $\vee$ is logical or, $f : \textbf{Bool} \to \textbf{Bool}$ is the identity, and $g : \textbf{Bool} \times \textbf{Bool} \to \mathbb{R}$:

$$g(x, y) = \begin{cases} 1 - c & \text{if } x = Y \\ 1 & \text{if } x = N \text{ and } y = Y \\ 1 - c + \epsilon & \text{if } x = N \text{ and } y = N \end{cases}$$

- The weakest-link public goods game (Example 6) has the monoid $\mathbb{R}_+^\infty = (\{\mathbb{R}_+ \cup \{\infty\}, \min, \infty)$, f the embedding $\mathbb{R}_+ \hookrightarrow \mathbb{R}_+^\infty$, and $g : \mathbb{R}_+ \times \mathbb{R}_+^\infty \to \mathbb{R}$ is $g(x, y) = \min(x, y) - c(x)$.

3 Open games

Open games were introduced in [13] as a compositional approach to game theory.

▶ **Definition 8** (Open game). *Let X, Y, R, S, Σ be sets. An open game $\mathcal{G}\colon \binom{X}{S} \overset{\Sigma}{\nrightarrow} \binom{Y}{R}$ has:*

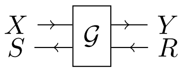
(i) *$\mathbb{P}_\mathcal{G} \colon \Sigma \times X \to Y$, called play function*

(ii) *$\mathbb{C}_\mathcal{G} \colon \Sigma \times X \times R \to S$, called coplay function*

(iii) *$\mathbb{B}_\mathcal{G} \colon X \times (Y \to R) \to \mathcal{P}(\Sigma^2)$, called best response function.*

Roughly speaking, an open game is a process that *(i)* given a *strategy* and *observation*, decides a *move*, and *(ii)* given a strategy, observation, and a *utility*, returns a *coutility* to the environment. Coutility is not a concept of classical game theory, but it enables compositionality by incorporating the fact that players reason about the future consequences of their actions. Finally, *(iii)*, the best response function, which, given a context for the game returns a relation on the set of strategies. A strategy σ is related to another strategy σ' if the latter is a best response to the former.

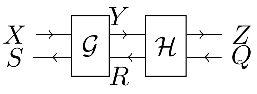
An open game is a process that receives observations (X) *"from the past"*, and the utility (R) *"from the future"*. It outputs moves (Y) covariantly and coutility (S) contravariantly.

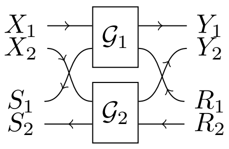


Open games are morphisms in a symmetric monoidal category **Game**. In order to formally define composition and monoidal product of games, it is useful to rephrase the definition in terms of lenses [19]. The detailed definitions are given in [13].

▶ **Definition 9 (Game). Game** *is the symmetric monoidal category with pairs of sets* $\left(\begin{smallmatrix}X\\S\end{smallmatrix}\right)$ *as objects and (equivalence classes of) open games* $\mathcal{G}\colon \left(\begin{smallmatrix}X\\S\end{smallmatrix}\right) \xrightarrow{\Sigma} \left(\begin{smallmatrix}Y\\R\end{smallmatrix}\right)$ *as morphisms.*

We give some intuitions. Composition, shown below left, is sequential play: $\mathcal{H} \cdot \mathcal{G}$ is thought of as $\mathcal{H}$ happening after $\mathcal{G}$, observing the moves of $\mathcal{G}$ and feeding back its coutility as $\mathcal{G}$'s utility. The monoidal product of open games represents two games played independently. The games are placed side by side with no connections, as shown below right.





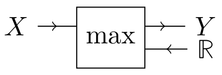
Classical games are *scalars* in **Game**, i.e. open games $\left(\begin{smallmatrix}1\\1\end{smallmatrix}\right) \nrightarrow \left(\begin{smallmatrix}1\\1\end{smallmatrix}\right)$. The fix-points of the best response functions of scalars in **Game** are the Nash equilibria of the games they represent.

Next we define specific open games used in our compositional account of network games. The first is the Utility Maximising Player, modelling typical players of classical game theory.

▶ **Definition 10** (Utility Maximising Player). *Let X and Y be sets and* $\mathsf{argmax}\colon \mathbb{R}^Y \to \mathcal{P}(Y)$ *take a function $\kappa : Y \to \mathbb{R}$ to the subset of Y where κ is maximised. Define $\mathcal{D}$ to be:*

$$\mathcal{D}\colon \left(\begin{smallmatrix}X\\1\end{smallmatrix}\right) \xrightarrow{Y^X} \left(\begin{smallmatrix}Y\\\mathbb{R}\end{smallmatrix}\right)$$

$$\begin{cases}\mathbb{P}_{\mathcal{D}}(f,x) = f(x)\\ \mathbb{C}_{\mathcal{D}}(f,x,r) = *\\ \mathbb{B}_{\mathcal{D}}(x,\kappa) = \{(y,y') \in Y^X \times Y^X : y'(x) \in \mathsf{argmax}(\kappa)\}\end{cases}$$

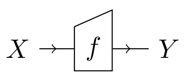


The category of sets and functions **Set** embeds into **Game** in two ways. In our account of network games, these embeddings encode how neighbours influence each other's utilities.

▶ **Definition 11.** *Let X, Y be sets and $f\colon X \to Y$ a function. Its covariant lifting is defined:*

$$f^*\colon \left(\begin{smallmatrix}X\\1\end{smallmatrix}\right) \xrightarrow{1} \left(\begin{smallmatrix}Y\\1\end{smallmatrix}\right)$$

$$\begin{cases}\mathbb{P}_{f^*}(*,x) = f(x)\\ \mathbb{C}_{f^*}(*,x,*) = *\\ \mathbb{B}_{f^*}(x,*) = \{(*,*)\}\end{cases}$$



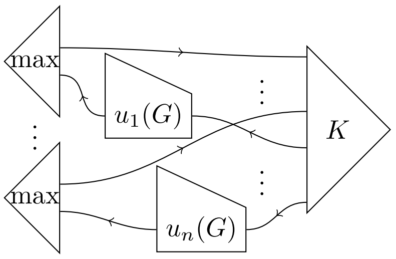
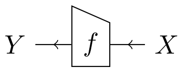


Figure 1 Open game representing a network game $\mathcal{N}$ played on a multigraph G

Similarly, its contravariant lifting is the following:

$$f_*\colon\ \left(\begin{smallmatrix}1\\Y\end{smallmatrix}\right) \overset{1}{\nrightarrow} \left(\begin{smallmatrix}1\\X\end{smallmatrix}\right)$$

$$\begin{cases} \mathbb{P}_{f_*}(*,*) = * \\ \mathbb{C}_{f_*}(*,*,x) = f(x) \\ \mathbb{B}_{f_*}(*,x) = \{(*,*)\} \end{cases}$$



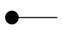
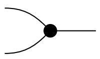
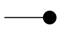
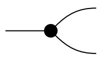
To obtain Examples 4, 5 and 6 as scalars in **Game**, players are taken to be utility-maximising players. The connectivity of the multigraph G determines their utility functions as contravariant liftings $u_i(G)$, while the context K sends back the choices of all players:

$$K\colon\ \left(\begin{smallmatrix}X^n\\X^n\times\cdots\times X^n\end{smallmatrix}\right) \overset{1}{\nrightarrow} \left(\begin{smallmatrix}1\\1\end{smallmatrix}\right)$$
$$\mathbb{C}_K(\underline{x}) = (\underline{x}, \ldots, \underline{x}).$$

The respective games are then obtained as the composition illustrated in Fig. 1. In this way, we obtain a compositional description of any network game. If a game can be described in the form of a a *monoid* network game, we can say more: such games are a monoidal functor from **Grph**, defined in the next section, to **Game**. The details are in Section 5.

4 Open graphs

We extend the compositional approach to graph theory of [7] from simple graphs to undirected multigraphs, identifying a "syntax" of network games as the arrows of a prop[1] **Grph**, generated from a monoidal signature and equations. We also provide a characterisation of **Grph** that explains its arrows as "open graphs". Differently from other approaches [3, 9], **Grph** uses adjacency matrices (Definition 2). Indeed, the presentation includes generators

 $: 0 \to 1$,  $: 2 \to 1$,  $: 1 \to 0$,  $: 1 \to 2$ (BIALG)

and the equations of Fig. 2. The prop **B** generated by this data is isomorphic [15, 24] to the prop of matrices with entries from $\mathbb{N}$, with composition being matrix multiplication. To convert between the two, think of the matrix as recording the numbers of paths: indeed, the (i,j)th entry in the matrix is the number of paths from the ith left port to the jth right port.

[1] A prop [17, 15] is a symmetric strict monoidal category where the objects are $\mathbb{N}$, and $m \otimes n := m + n$.

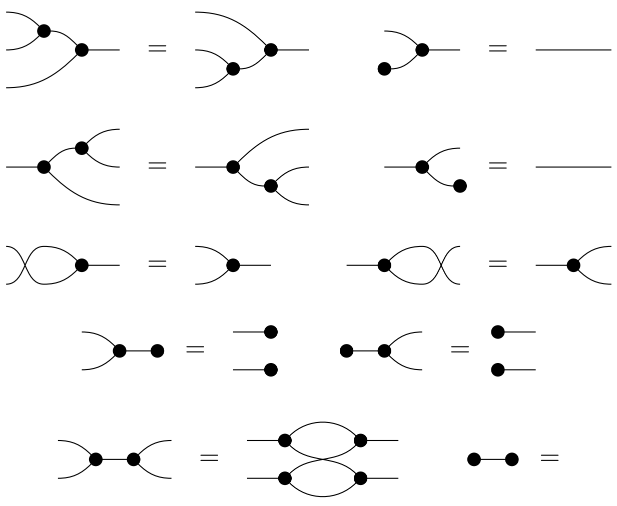


Figure 2 Commutative bialgebra equations, yielding prop **B**.

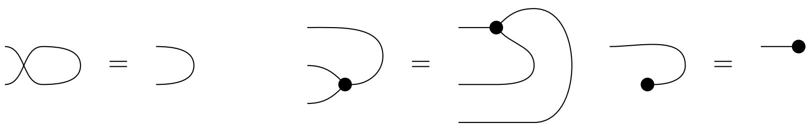
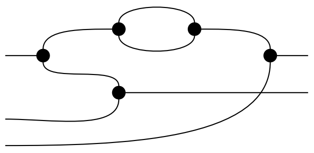
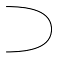


Figure 3 Equations of ∪, which together with the equations of Fig. 2 yield prop **BU**.

▶ **Example 12.** The following string diagram in **B** corresponds to the 3×2 matrix $\left(\begin{smallmatrix} 2 & 1 \\ 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$.



Next, we add a "cup" generator denoted

 $: 2 \to 0$ (U)

with its equations given in Fig. 3.

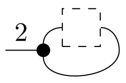
▶ **Definition 13.** *Let* **BU** *be the prop obtained from* (BIALG) *and* (U)*, quotiented by equations in Figs. 2 and 3, where the empty diagram is the identity on the monoidal unit.*

Just as **B** captures ordinary matrices, **BU** captures adjacency matrices:

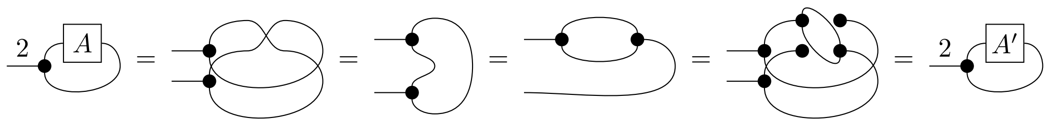
▶ **Proposition 14.** *For $n \in \mathbb{N}$, the hom-set $[n, 0]$ of* **BU** *is in bijection with $n \times n$ adjacency matrices, in the sense of Definition 2.*

We have seen that the relationship between matrices and diagrams in **B** is that the former encode the path information from the latter. Thus an $m \times n$ matrix is a diagram from m to n. Adding the cup and the additional equations means that, in general, a diagram from n to 0 in **BU** "encapsulates" an $n \times n$ matrix that expresses connectivity information in a similar way to adjacency matrices. We now give a concrete derivation to demonstrate this.

▶ **Example 15.** The equivalence relation of adjacency matrices is captured by the equations of Fig. 3. Consider matrices $A = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right) \sim \left(\begin{smallmatrix} 0 & 2 \\ 0 & 0 \end{smallmatrix}\right) = A'$. The morphism in **BU** is obtained by constructing their diagram in **B** as in Example 12 and "plugging" them in the following.

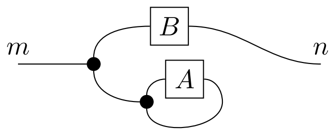


As shown below, the two diagrams obtained are equated by the axioms of **BU**.



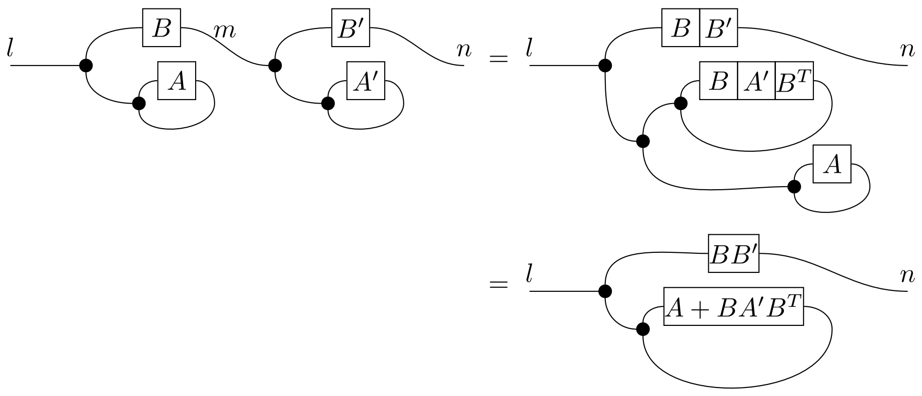
The prop **BU** can be given a straightforward combinatorial characterisation as the prop **Adj**.

▶ **Definition 16 (Adj).** *A morphism $\alpha\colon m \to n$ in the prop* **Adj** *[7] is a pair $(B, [A])$, where $B \in \mathbf{Mat}_{\mathbb{N}}(m, n)$ is a matrix, while $[A]$, with $A \in \mathbf{Mat}_{\mathbb{N}}(m, m)$, is an adjacency matrix. The components of* **Adj** *morphisms can be read off a "normal form" for* **BU** *arrows, as follows.*



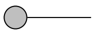
Composition in **Adj** becomes intuitive when visualised with string diagrams.

$(B, [A]) \circ (B', [A']) = (BB', [A + BA'B^T])$



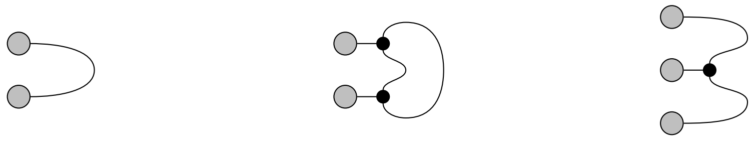
▶ **Proposition 17.** **BU** *is isomorphic to the prop* **Adj**. ◀

The proof is similar to the case for $\mathbb{Z}_2$ [7]. An extension of **BU** with just one additional generator and no additional equations yields the prop **Grph** of central interest for us.

▶ **Definition 18.** *The prop* **Grph** *is obtained from the generators in* (BIALG) *and* (U) *together with a generator*  $: 0 \to 1$. *The equations are those of Figs. 2 and 3.*

As we shall see, arrows $0 \to 0$ in **Grph** are precisely finite undirected multigraphs taken up to isomorphism: the additional generator plays the role of a graph vertex.

▶ **Example 19.** For example, the first of the following represents a multigraph with two vertices, connected by a single edge. The second one, two vertices connected by two edges. The third one, is a multigraph with three vertices and two edges between them.



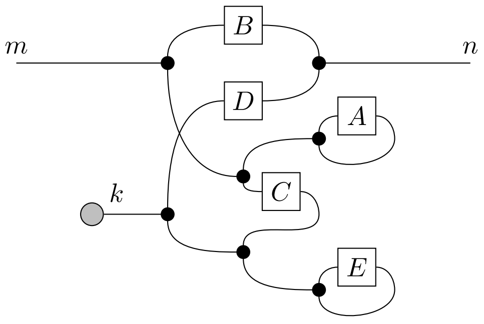
While the arrows $[0,0]$ are (iso classes of) multigraphs, general arrows can be understood as open graphs. Roughly speaking, they are graphs together with interfaces, and data that specifies the connectivity of the graph to its interfaces. We make this explicit below. Indeed, we shall see (Theorem 23) that the prop $\mathcal{A}$, defined below, is isomorphic to **Grph** – for this reason we use **Grph** string diagrams to illustrate its structure.

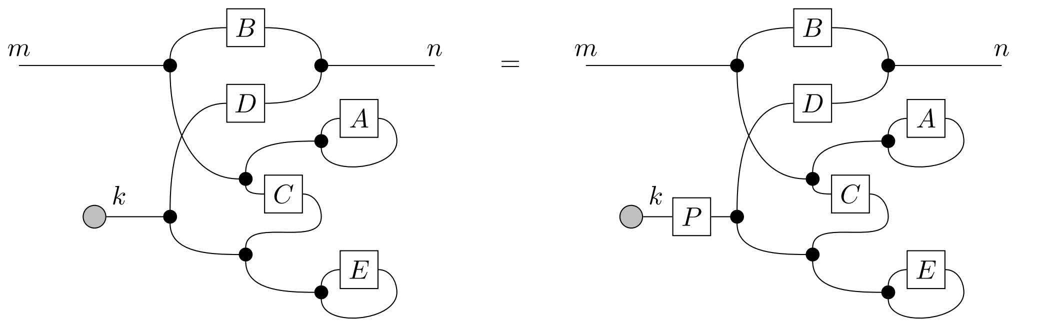
▶ **Definition 20** (The prop $\mathcal{A}$). *A morphism $\Gamma\colon m \to n$ in the prop $\mathcal{A}$ is defined by*

$$\Gamma = (k, [A], B, C, D, [E]) \tag{3}$$

where $k \in \mathbb{N}$, $A \in \mathbf{Mat}_{\mathbb{N}}(m,m)$, $B \in \mathbf{Mat}_{\mathbb{N}}(m,n)$, $C \in \mathbf{Mat}_{\mathbb{N}}(m,k)$, $D \in \mathbf{Mat}_{\mathbb{N}}(k,n)$ and $E \in \mathbf{Mat}_{\mathbb{N}}(k,k)$. Similarly to **Adj** *(Definition 16), the components of* (3) *can be read off a "normal form" for arrows of* **Grph**, *as visualised below right.*

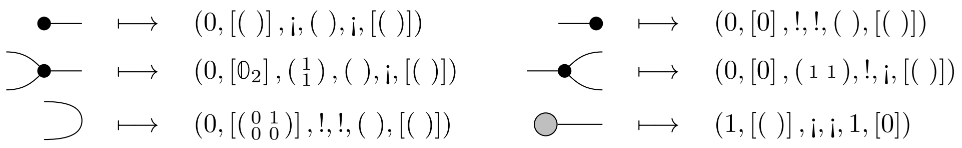
Tuples (3) *are taken up to an equivalence relation that captures the fact that the order of the vertices is immaterial. Let $\Gamma \sim \Gamma'$ iff they are morphisms of the same type, $\Gamma, \Gamma'\colon m \to n$ with k vertices, and there is a permutation matrix $P \in \mathbf{Mat}(k,k)$ such that $\Gamma' = (k, [A], B, CP^T, PD, [PEP^T])$. The justification for this equivalence is the equality of the following two string diagrams in* **Grph**, *below (for the details, see Appendix A on page 18).*





It is worthwhile to give some intuition for the components of (3). The idea is that an arrow Γ specifies a multigraph $G = (k, [E])$, and:

- B specifies connections between the two boundaries, bypassing G
- C specifies connections between the left boundary and G
- D specifies connections between G and the right boundary
- A specifies connections between the interfaces on the left boundary. This allows Γ to introduce connections between the vertices of an "earlier" open graph Δ. See Example 21.

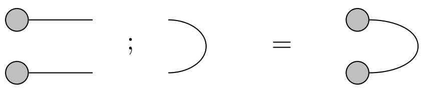


$$\mapsto \quad (0, [(\ )]\,, ¡, (\ ), ¡, [(\ )]) \qquad \mapsto \quad (0, [0]\,, !, !, (\ ), [(\ )])$$
$$\mapsto \quad (0, [\mathbb{0}_2]\,, (\begin{smallmatrix}1\\1\end{smallmatrix}), (\ ), ¡, [(\ )]) \qquad \mapsto \quad (0, [0]\,, (\begin{smallmatrix}1 & 1\end{smallmatrix}), !, ¡, [(\ )])$$
$$\mapsto \quad (0, [(\begin{smallmatrix}0 & 1\\0 & 0\end{smallmatrix})]\,, !, !, (\ ), [(\ )]) \qquad \mapsto \quad (1, [(\ )]\,, ¡, ¡, 1, [0])$$

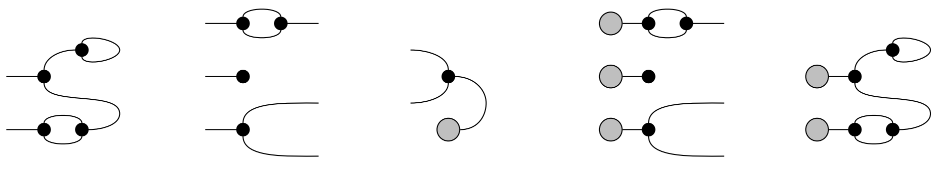
Figure 4 Image of θ on the generators

Defining composition in $\mathcal{A}$ is straightforward, given the above intuitions, but the details are rather tedious: see Lemma 33 in Appendix A.

▶ **Example 21.** In a composite $\Delta\ ;\Gamma$, Γ may introduce edges between the vertices of Δ. Indeed, the first diagram in Example 19 can be decomposed:



▶ **Example 22.** The following show the role of $\mathcal{A}$-morphism components, when isolated. The leftmost open graph has only left-side ports. It introduces a self-loop and two connections. The second has only connections between the left and right interfaces; the first left port is connected twice to the first right port, the second port is disconnected, and the third left port is connected to the second and third right ports. The third open graph has one vertex connected to the two left ports. The fourth has three vertices connected to the right ports, following the specification in the second. The rightmost (closed) multigraph has its vertices connected according to the specification of the leftmost vertex-less open graph. We write ! for matrices without columns, ¡ for matrices without rows and ( ) for the empty matrix.



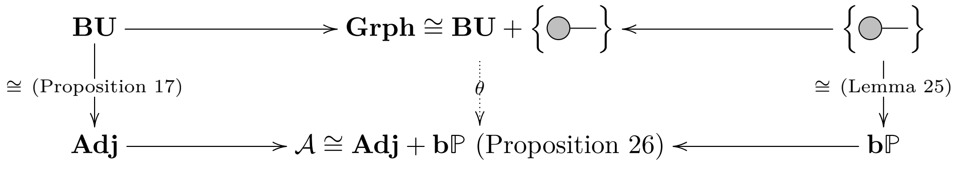
$(0,[A],!,!,¡,[(\ )])$ $\quad(0,[\mathbb{0}],B,!,¡,[(\ )])$ $\quad(1,[\mathbb{0}],!,C,!,[0])$ $\quad(3,[(\ )],¡,¡,D,[\mathbb{0}])$ $\quad(2,[(\ )],(\ ),¡,!,[E])$

$A=\begin{pmatrix}1&0\\2&0\end{pmatrix}$ $\quad B=\begin{pmatrix}2&0&0\\0&0&0\\0&1&1\end{pmatrix}$ $\quad C=\begin{pmatrix}1\\1\end{pmatrix}$ $\quad D=\begin{pmatrix}2&0&0\\0&0&0\\0&1&1\end{pmatrix}$ $\quad E=\begin{pmatrix}1&0\\2&0\end{pmatrix}$

The main result in this section is the following.

▶ **Theorem 23.** *There is an isomorphism of props* $\theta\colon \mathbf{Grph} \to \mathcal{A}$.

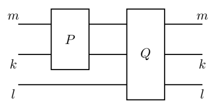
The remainder of this section builds a proof of the above, summarised in the diagram below.



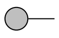
First, note that **Grph** is the coproduct $\mathbf{BU}+\{\text{●—}\}$ in the category of props, where $\{\text{●—}\}$ is the free prop on a single generator $0 \to 1$. Next, we characterise $\{\text{●—}\}$ as $\mathbf{b}\mathbb{P}$, defined below, in Lemma 25. Given that $\mathbf{BU} \cong \mathbf{Adj}$, as shown in Proposition 17, to show the existence of θ it suffices to show that $\mathcal{A}$ satisfies the universal property of the coproduct $\mathbf{Adj} + \mathbf{b}\mathbb{P}$, which is Proposition 26. The action of θ on the generators of $\mathbf{Grph}$ is in Fig. 4.

▶ **Definition 24** ($\mathbf{b}\mathbb{P}$). *The prop of* bound permutations $\mathbf{b}\mathbb{P}$ *has as morphisms $m \to m+k$ pairs $[(k,P)]$ where $k \in \mathbb{N}$ and $P \in \mathbf{Mat}_{\mathbb{N}}(m+k, m+k)$ is a permutation matrix. Such pairs are identified to ensure that the order of the lower k rows of P is immaterial. Roughly speaking, considering P as a permutation of $m+k$ inputs to $m+k$ outputs, in $[(k,P)]$ the final k inputs are "bound". Explicitly, $(k,P) \sim (k,P')$ iff there is a permutation $\sigma \in \mathbf{Mat}_{\mathbb{N}}(k,k)$ st $P = \left(\begin{smallmatrix} \mathbb{1}_m & \mathbb{0} \\ \mathbb{0} & \sigma \end{smallmatrix}\right) P'$. Composition is defined:*

$$(l,Q) \circ (k,P) = (k+l, \left(\begin{smallmatrix} P & \mathbb{0} \\ \mathbb{0} & \mathbb{1}_l \end{smallmatrix}\right) Q)$$



Identities are identity matrices $id_n = (0, \mathbb{1}_n)$. The fact that $\mathbf{b}\mathbb{P}$ is a prop is Lemma 34 in Appendix A.

▶ **Lemma 25.** $\mathbf{b}\mathbb{P}$ *is isomorphic to* $\{$$\}$.

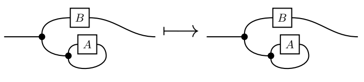
Proof. Let us call $\phi = (0,(1)) \colon 0 \to 1$, which is a morphism in $\mathbf{b}\mathbb{P}$. We show directly that, for any other prop $\mathbb{P}$ that contains a morphism $v \colon 0 \to 1$, there is a unique prop homomorphism $\alpha^{\#} \colon \mathbf{b}\mathbb{P} \to \mathbb{P}$ such that $\alpha^{\#}(\phi) = v$. The details are given as Lemma 35 in Appendix A. ◀

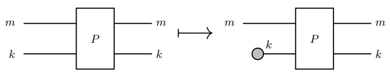
Given the results of Proposition 17 and Lemma 25, we obtain the isomorphism $\theta \colon \mathbf{Grph} \to \mathcal{A}$, thereby completing the proof of Theorem 23, by showing that:

▶ **Proposition 26.** $\mathcal{A}$ *satisfies the universal property of the coproduct* $\mathbf{Adj} + \mathbf{b}\mathbb{P}$.

Proof. In order to show that $\mathcal{A}$ is a coproduct $\mathbf{Adj} + \mathbf{b}\mathbb{P}$, we define the two inclusions.

$$\begin{aligned} i_1 \colon \mathbf{Adj} &\longrightarrow \mathcal{A} \\ n &\longmapsto n \\ (B,[A]) &\longmapsto (0,[A],B,!,¡,[(\ )]) \end{aligned} \qquad \begin{aligned} i_2 \colon \mathbf{b}\mathbb{P} &\longrightarrow \mathcal{A} \\ n &\longmapsto n \\ (k,P) &\longmapsto (k,[\mathbb{0}_n],P_*^{[1,n]},\mathbb{0}_{nk},P_*^{[n+1,n+k]},[\mathbb{0}_k]) \end{aligned}$$





We indicate with $P_*^{[1,n]}$ the first n rows of the matrix P and, similarly, with $P_*^{[n+1,n+k]}$ the rows between the $n+1$-th and the $n+k$-th. It is not difficult to show that these are indeed homomorphism, the details are given as Claim 36 in Appendix A.

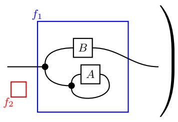
Now, we show that, for any other prop $\mathcal{C}$ with prop homomorphisms $\mathbf{Adj} \xrightarrow{f_1} \mathcal{C} \xleftarrow{f_2} \mathbf{b}\mathbb{P}$, there exists a unique prop homomorphism $H \colon \mathcal{A} \to \mathcal{C}$ such that $H \circ i_1 = f_1$ and $H \circ i_2 = f_2$.

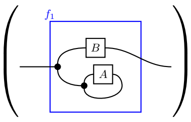
$$\begin{aligned} H \colon \mathcal{A} &\longrightarrow \mathcal{C} \\ n &\longmapsto n \\ (k,[A],B,C,D,[E]) &\longmapsto f_1\left(\left(\begin{smallmatrix} B \\ D \end{smallmatrix}\right), \left[\left(\begin{smallmatrix} A & C \\ \mathbb{0} & E \end{smallmatrix}\right)\right]\right) \circ (\mathbb{1}_m \otimes f_2(k, \mathbb{1}_k)) \end{aligned}$$

We verify that H is a homomorphism in Lemma 37 in Appendix A. Next, we confirm that $H \circ i_1 = f_1$ and $H \circ i_2 = f_2$, where two functor boxes [8] for f_1 and f_2 are coloured:

$$H \circ i_1(B, [A]) = H(0, [A], B, !, ¡, [(\ )])$$

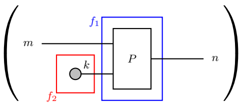
$$= f_1(B, [A]) \circ (\mathbb{1}_m \otimes f_2(0, \mathbb{1}_0)) \left(\quad \right) = f_1(B, [A]) \left(\quad \right)$$

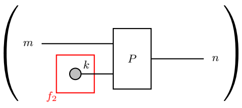




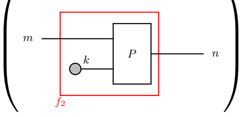
$$H \circ i_2(k, P) = H(k, [\mathbb{0}_n], P_*^{[1,n]}, \mathbb{0}_{nk}, P_*^{[n+1,n+k]}, [\mathbb{0}_k])$$

$$= f_1(P, [\mathbb{0}_{n+k}]) \circ (\mathbb{1}_n \otimes f_2(k, \mathbb{1}_k)) \left(\quad \right) = P \circ f_2(k, \mathbb{1}_{n+k}) \left(\quad \right)$$





$$= f_2(k, P) \left(\quad \right)$$



Moreover, H is the unique prop homomorphism with these properties. In fact, suppose there is $H' \colon \mathcal{A} \to C$ such that $H' \circ i_1 = f_1$ and $H' \circ i_2 = f_2$. Then:

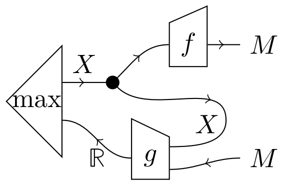
$$\begin{aligned} H'(k, [A], B, C, D, [E]) &= H'(i_1((\begin{smallmatrix} B \\ D \end{smallmatrix}), [(\begin{smallmatrix} A & C \\ \mathbb{0} & E \end{smallmatrix})]) \circ (\mathbb{1}_m \otimes i_2(k, \mathbb{1}_k))) \\ &= H' i_1((\begin{smallmatrix} B \\ D \end{smallmatrix}), [(\begin{smallmatrix} A & C \\ \mathbb{0} & E \end{smallmatrix})]) \circ (H'(\mathbb{1}_m) \otimes H' i_2(k, \mathbb{1}_k)) \\ &= f_1((\begin{smallmatrix} B \\ D \end{smallmatrix}), [(\begin{smallmatrix} A & C \\ \mathbb{0} & E \end{smallmatrix})]) \circ (\mathbb{1}_m \otimes f_2(k, \mathbb{1}_k)) = H(k, [A], B, C, D, [E]). \blacktriangleleft \end{aligned}$$

5 Games on graphs via functorial semantics

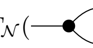
Here we show that monoid network games $\mathcal{N}$ define monoidal functors $F_{\mathcal{N}} \colon \mathbf{Grph} \to \mathbf{Game}$, which is our main contribution. To every open graph Γ, $F_{\mathcal{N}}$ associates an open game, where $\mathcal{N}$ is played on Γ. We give an explicit account of the $F_{\mathcal{N}}$-image of open graphs Γ, using Theorem 23. We also explain how $F_{\mathcal{N}}$ acts on closed graphs, giving classical games.

Since **Grph** is given by generators and equations, it suffices to define $F_{\mathcal{N}}$ on the generators and show that the equations are respected. Fix a monoid network game $\mathcal{N} = (X, M, f, g)$.

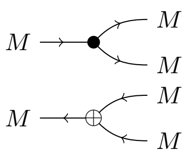
- On objects, $F_{\mathcal{N}}(1) = (\begin{smallmatrix} M \\ M \end{smallmatrix})$. Thus, for $n \in \mathbf{Grph}$, we have $F_{\mathcal{N}}(n) = (\begin{smallmatrix} M^n \\ M^n \end{smallmatrix})$
- The vertex ●— $: 0 \to 1$ is mapped to the open game $F_{\mathcal{N}}($●—$) \colon (\begin{smallmatrix} 1 \\ 1 \end{smallmatrix}) \overset{X}{\nrightarrow} (\begin{smallmatrix} M \\ M \end{smallmatrix})$ defined
  - $\Sigma_{F_{\mathcal{N}}(\bullet\!-)} = X$
  - $\mathbb{P}_{F_{\mathcal{N}}(\bullet\!-)}(x_i, *) = f(x_i)$
  - $\mathbb{C}_{F_{\mathcal{N}}(\bullet\!-)}(x_i, *, m) = *$
  - $(x_i, x_i') \in \mathbb{B}_{F_{\mathcal{N}}(\bullet\!-)}(*, \kappa : M \to M)$
    iff $x_i' \in \arg\max\limits_{x_i'' : X} g(x_i'', \kappa(f(x_i'')))$

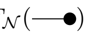


- The generators (BIALG) are mapped to the bialgebra structure on (M, M) induced by the monoid action of M. Specifically, they are:

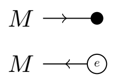
$F_{\mathcal{N}}($$)\colon \left(\begin{smallmatrix} M \\ M \end{smallmatrix}\right) \xrightarrow{1} \left(\begin{smallmatrix} M^2 \\ M^2 \end{smallmatrix}\right)$

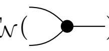
$$\begin{cases} \mathbb{P}(*, m) = (m, m) \\ \mathbb{C}(*, m_1, m_2, m_3) = m_2 \oplus m_3 \end{cases}$$



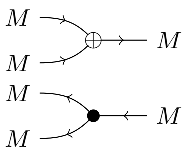
$F_{\mathcal{N}}($$)\colon \left(\begin{smallmatrix} M \\ M \end{smallmatrix}\right) \xrightarrow{1} \left(\begin{smallmatrix} 1 \\ 1 \end{smallmatrix}\right)$

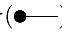
$$\begin{cases} \mathbb{P}(*, m) = * \\ \mathbb{C}(*, m, *) = e \end{cases}$$



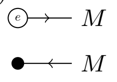
$F_{\mathcal{N}}($$)\colon \left(\begin{smallmatrix} M^2 \\ M^2 \end{smallmatrix}\right) \xrightarrow{1} \left(\begin{smallmatrix} M \\ M \end{smallmatrix}\right)$

$$\begin{cases} \mathbb{P}(*, m_1, m_2) = m_1 \oplus m_2 \\ \mathbb{C}(*, m_1, m_2, m_3) = (m_1, m_1) \end{cases}$$

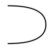


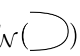
$F_{\mathcal{N}}($$)\colon \left(\begin{smallmatrix} 1 \\ 1 \end{smallmatrix}\right) \xrightarrow{1} \left(\begin{smallmatrix} M \\ M \end{smallmatrix}\right)$

$$\begin{cases} \mathbb{P}(*, *) = e \\ \mathbb{C}(*, *, m) = * \end{cases}$$

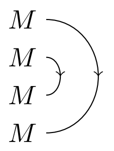


where each of these open games is built from lifted functions (Definition 11).

- : $2 \to 0$ is mapped to the following open game (see [13])

$F_{\mathcal{N}}($$)\colon \left(\begin{smallmatrix} M^2 \\ M^2 \end{smallmatrix}\right) \xrightarrow{1} \left(\begin{smallmatrix} 1 \\ 1 \end{smallmatrix}\right)$

$$\begin{cases} \mathbb{P}(*, m_1, m_2) = * \\ \mathbb{C}(*, m_1, m_2, *) = (m_2, m_1) \end{cases}$$



To prove that $F_{\mathcal{N}}$ is a symmetric monoidal functor it suffices to show that the equations of **Grph** are respected; this is a straightforward but somewhat lengthy computation.

▶ **Theorem 27.** *$F_{\mathcal{N}}$ defines a symmetric monoidal functor* **Grph** $\to$ **Game***.*

Proof. See Appendix B, on page 23. ◀

Note that $F_{\mathcal{N}}$ does *not* respect axioms (C1) or (C2) of [7], so it does not define a functor **ABUV** $\to$ **Game** in the terminology of *loc. cit.* This, together with the increased expressivity of multigraphs over simple graphs, motivates our extension from **ABUV** to **Grph**.

Theorem 23 gives a convenient "normal form" for the arrows of **Grph**, which we use to give an explicit description of the image of any (open) graph Γ under $F_{\mathcal{N}}$. First, we specialise to closed graphs that yield ordinary network games. This result—a sanity check for our compositional framework—is a corollary of the more general Theorem 29, proved subsequently.

▶ **Corollary 28.** *Let $\mathcal{N} = (X, M, f, g)$ be a monoid network game, and consider $\Gamma : 0 \to 0$ in* **Grph***, an undirected multigraph with k vertices. Then the game $F_{\mathcal{N}}(\Gamma)\colon \left(\begin{smallmatrix} 1 \\ 1 \end{smallmatrix}\right) \xrightarrow{X^k} \left(\begin{smallmatrix} 1 \\ 1 \end{smallmatrix}\right)$ has:*
- *$\Sigma_{F_{\mathcal{N}}(\Gamma)} = X^k$ as its strategy profiles,*
- *$\mathbb{B}_{F_{\mathcal{N}}(\Gamma)}(*, *) \subseteq X^k \times X^k$ is the best response relation of $\mathcal{N}$ played on Γ.*

Note that while the expressions in the statement of Theorem 29 below may seem involved, they are actually derived in an entirely principled, compositional manner from the generators of **Grph**. Indeed, the proof is by structural induction on the morphisms of **Grph**.

▶ **Theorem 29.** *Let $\mathcal{N} = (X, M, f, g)$ be a monoid network game. Let $\Gamma : i \to j$ be a morphism in* **Grph** *with k vertices st $\theta(\Gamma) = (k, [A], B, C, D, [E])$, where $A : i \times i$, $B : i \times j$, $C : i \times k$, $D : k \times j$ and $E : k \times k$. Then the open game $F_{\mathcal{N}}(\Gamma)\colon \left(\begin{smallmatrix} M^i \\ M^i \end{smallmatrix}\right) \xrightarrow{X^k} \left(\begin{smallmatrix} M^j \\ M^j \end{smallmatrix}\right)$ has:*
- *set of strategy profiles $\Sigma(F_{\mathcal{N}}(\Gamma)) = X^k$*
- *play function $\mathbb{P}_{F_{\mathcal{N}}(\Gamma)} : X^k \times M^i \to M^j$ given by $\mathbb{P}_{F_{\mathcal{N}}(\Gamma)}(\underline{\sigma}, \underline{x}) = B^T \underline{x} \oplus D^T f(\underline{\sigma})$*
- *coplay function $\mathbb{C}_{F_{\mathcal{N}}(\Gamma)} : X^k \times M^i \times M^j \to M^i$ is $\mathbb{C}_{F_{\mathcal{N}}(\Gamma)}(\underline{\sigma}, \underline{x}, \underline{r}) = (A + A^T)\underline{x} \oplus B\underline{r} \oplus C f(\underline{\sigma})$*

- *best response relation* $\mathbb{B}_{F_\mathcal{N}(\Gamma)} : M^i \times (M^j \to M^j) \to \mathcal{P}(X^k \times X^k)$ *is* $(\underline{\sigma}, \underline{\sigma}') \in \mathbb{B}_{F_\mathcal{N}(\Gamma)}(\underline{x}, \kappa)$ *iff, for all* k*,*

$$\sigma'_k \in \operatorname*{argmax}_{s \in X} g\left(s, (C^T)^k_* \underline{x} \oplus D^k_* \kappa \left(B^T \underline{x} \oplus D^T f(\underline{\sigma}\,[k \mapsto s])\right) \oplus (E + E^T)^k_* f(\underline{\sigma}\,[k \mapsto s])\right)$$

Proof. See Appendix B on page 23. ◀

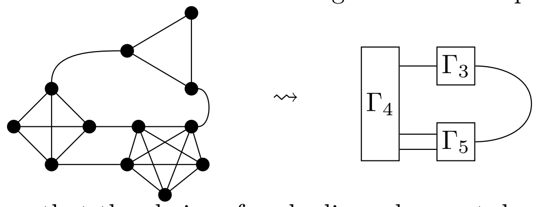
6 Examples

We return to examples: the majority (Example 4), the best-shot public goods (Example 5) and the weakest-link public goods (Example 6) games, and demonstrate various applications of our framework. We first show that to compute the Nash equilibrium of the majority game played on interconnected cliques is to calculate equilibria of its clique subgames.

▶ **Example 30** (Majority game). In the majority game the best response can be decomposed into the best responses of its components. Let $\mathcal{N}$ be the monoid network game for the majority game, defined on pg. 5, and consider a graph composed of N cliques, as follows:

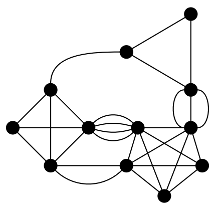
- each vertex of each clique can be connected to at most one vertex of another clique,
- in each clique there is at least one vertex not connected to any vertex outside its clique.

Such graphs decompose as N open graphs, each a clique with some boundary connections. We omit the details and give, instead, an illustrative example: below left is a picture of three connected cliques, while the schematic on the right is the corresponding expression in **Grph**.



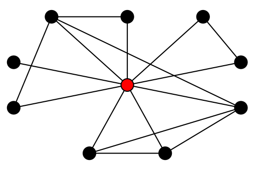
It is easy to show that the choice of each clique does not depend on the choices of other cliques. Indeed, the Nash equilibria of the majority game played on connected cliques in our sense are those strategy profiles where, in every clique, all players make the same choice. In particular, there are 2^N Nash equilibria.

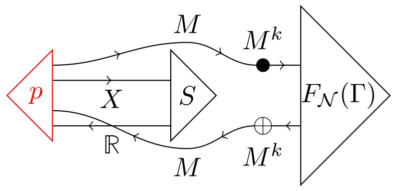
In some cases, players can take into account the choice of another player with a different intensity. This can be modelled by changing the number of edges between the vertices. Let us consider the above example with some of the vertices connected multiple times. This modification of the network—illustrated below—reflects in a modification of the equilibria, which are now strategy profiles in which every player takes the same choice.



In the best-shot public goods game (Example 5), the Nash equilibrium is when no player invests. In Example 31, we show how the compositional description is useful to adapt the model to a slightly different situation. We can imagine that one of the players now has access to incentives to invest in the public good. This scenario is represented by modifying the game and allowing one player to interact with the environment, which is the source of the incentives for this player. This modification "opens" the game to one of type $\left(\begin{smallmatrix}1\\1\end{smallmatrix}\right) \to \left(\begin{smallmatrix}X\\\mathbb{R}\end{smallmatrix}\right)$: as a result, the Nash equilibrium changes. This is a simple model of a common economic situation, 'solving' a social dilemma by external intervention, for example by regulation [12].

▶ **Example 31** (Best-shot public goods game). Consider the best-shot public goods game played on a graph that contains a vertex connected to all other vertices. Removing the central vertex from this graph leaves an open graph that we will call Γ.





Here, $F_{\mathcal{N}}(\Gamma)$ is the best-shot public goods game played on the open graph Γ, p is the central player that has been substituted, and S is the external open game that influences p. The utility function of player p and the coplay function of S are as follows.

$$u_p(\Gamma, x) = \begin{cases} 1 - c + \delta & \text{if } x_p = 1 \\ 1 - \epsilon & \text{if } x_p = 0 \wedge \exists (p, j) \in E_\Gamma \; x_j = 1 \\ 1 - c & \text{if } \forall j \; x_j = 0 \end{cases}$$

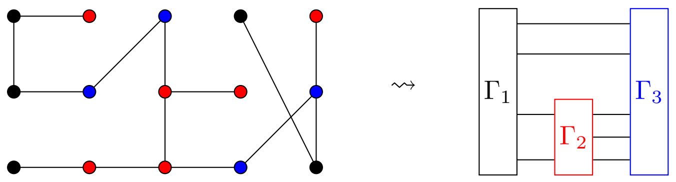
$$S\colon \; \left(\begin{smallmatrix} X \\ \mathbb{R} \end{smallmatrix}\right) \overset{1}{\nrightarrow} \left(\begin{smallmatrix} 1 \\ 1 \end{smallmatrix}\right)$$

$$\mathbb{C}(*, x, *) = \begin{cases} \delta & \text{if } x = 1 \\ -\epsilon & \text{if } x = 0 \end{cases}$$

The addition of the open game S and the modification of player p modifies the Nash equilibrium to be the strategy profile where only the central player invests. The idea is that the "external" agent S incentivises the central player p to invest.

Our last example illustrates a common situation where the compositional description of a game does *not* allow a compositional analysis of the best response. However, in this case, compositionality can be used to obtain a variant of the weakest-link public goods game (Example 6) where different cost functions are used in different parts of the graph G. The desired game is obtained by composing such open games according to the structure of G.

▶ **Example 32** (Weakest-link public goods game). Consider the weakest-link public goods game played on a connected graph G. Suppose that players have different cost functions. We partition them according to their cost functions, and use this partition to decompose the G into an expression in **Grph**, as illustrated for a particular example below:



While the definition above uses our compositional techniques, the Nash equilibrium is calculated on the resulting closed game, and is a strategy profile where every player invests equally, with utility depending on individual cost functions. While it may be unsatisfying, this failure of Nash equilibria to be compositional can be seen as an inherent feature of game theory. In particular it is already present in the theory of open games; the passage from graphs to games is nevertheless fully compositional.

7 Conclusions

Our contribution is a compositional account of network games via strict monoidal functors. This adds a class of network games to the games that have been expressed in compositional game theory [13, 2]. Of independent interest is our work on the category **Grph**, extending [7]. This is an approach to "open graphs" that, as we have seen, is compatible with the structure of open games, and in future work we will identify other uses of this category.

We also intend to extend the class of open graphs to *directed* open graphs. The motivation for this is that, in some network games, interactions between players are *not* bidirectional. Consider, for example, a variant of the majority game where there is an "influencer": a player whose choice affects the choices of other players, but is not in turn conversely affected.

We will also extend the menagerie of games that can be played on a graph. We plan to study games with more generic utility functions, incomplete information, and repeated games. It could also prove interesting to study natural transformations between the functors that define games, and explore the game theoretical relevance of such transformations.

References

1 John C Baez and Brendan Fong. A compositional framework for passive linear networks, 2015. arXiv preprint: `https://arxiv.org/abs/1504.05625`.

2 Joe Bolt, Jules Hedges, and Philipp Zahn. Bayesian open games, 2019. arXiv preprint: `https://arxiv.org/abs/1910.03656`.

3 Filippo Bonchi, Fabio Gadducci, Aleks Kissinger, Pawel Sobocinski, and Fabio Zanasi. Rewriting modulo symmetric monoidal structure. In Martin Grohe, Eric Koskinen, and Natarajan Shankar, editors, *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16, New York, NY, USA, July 5-8, 2016*, pages 710–719. ACM, 2016. `doi:10.1145/2933575.2935316`.

4 Filippo Bonchi, Pawel Sobocinski, and Fabio Zanasi. The calculus of signal flow diagrams I: linear relations on streams. *Inf. Comput.*, 252:2–29, 2017. `doi:10.1016/j.ic.2016.03.002`.

5 Yann Bramoullé, Andrea Galeotti, and Brian Rogers. *The Oxford handbook of the economics of networks*. Oxford University Press, 2016.

6 Roberto Bruni, Hernán C. Melgratti, and Ugo Montanari. A connector algebra for P/T nets interactions. In *Concurrency Theory (CONCUR '11)*, volume 6901 of *LNCS*, pages 312–326. Springer, 2011. `doi:10.1007/978-3-642-23217-6_21`.

7 Apiwat Chantawibul and Paweł Sobociński. Towards compositional graph theory. In *Proceedings of MFPS'15*, volume 319 of *ENTCS*, pages 121–136, 2015. `doi:10.1016/j.entcs.2015.12.009`.

8 J. R. B. Cockett and R. A. G. Seely. Linearly distributive functors. *Journal of Pure and Applied Algebra*, 143(1-3):155–203, 1999.

9 Brendan Fong. Decorated cospans. *Theory and Applications of Categories*, 30(33):1096–1120, 2015.

10 James H Fowler and Nicholas A Christakis. Dynamic spread of happiness in a large social network: longitudinal analysis over 20 years in the framingham heart study. *BMJ*, 337, 2008. `doi:10.1136/bmj.a2338`.

11 Andrea Galeotti. Talking, searching and pricing. *International Economic Review*, 51(4):1159–1174, 2010. `doi:10.1111/j.1468-2354.2010.00614.x`.

12 Andrea Galeotti, Benjamin Golub, and Sanjeev Goyal. Targeting interventions in networks. Forthcoming in *Econometrica*, 2019.

13 Neil Ghani, Jules Hedges, Viktor Winschel, and Philipp Zahn. Compositional game theory. In *Proceedings of Logic in Computer Science (LiCS) 2018*. ACM, 2018. `doi:10.1145/3209108.3209165`.

14 Matthew Jackson and Yves Zenou. Games on networks. In *Handbook of game theory with economic applications*, volume 4, chapter 3, pages 95–163. Elsevier, 2015. `doi:10.1016/B978-0-444-53766-9.00003-3`.

15 Stephen Lack. Composing PROPs. *Theor. App. Categories*, 13(9):147–163, 2004.

16 Qiu Li, MingChu Li, Lin Lv, Cheng Guo, and Kun Lu. A new prediction model of infectious diseases with vaccination strategies based on evolutionary game theory. *Chaos, Solitons & Fractals*, 104:51–60, 2017. `doi:10.1016/j.chaos.2017.07.022`.

17 Saunders Mac Lane. Categorical algebra. *Bull. Amer. Math. Soc.*, 71:40–106, 1965.

18 Saunders Mac Lane. *Categories for the Working Mathematician.* Springer-Verlag New York, 1978.

19 Frank Joseph Oles. *A Category-theoretic Approach to the Semantics of Programming Languages.* PhD thesis, Syracuse University, Syracuse, NY, USA, 1982. AAI8301650.

20 Alfred Tarski. The concept of truth in the languages of the deductive sciences. *Prace Towarzystwa Naukowego Warszawskiego, Wydzial III Nauk Matematyczno-Fizycznych*, 34(13-172):198, 1933.

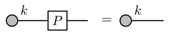
21 Alfred Tarski and Robert L Vaught. Arithmetical extensions of relational systems. *Compositio mathematica*, 13:81–102, 1957.

22 Giorgio Topa. Social Interactions, Local Spillovers and Unemployment. *The Review of Economic Studies*, 68(2):261–295, April 2001. doi:10.1111/1467-937X.00169.

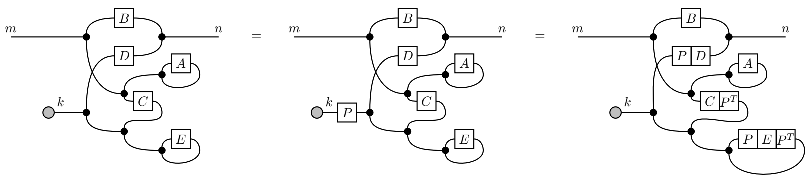
23 Glynn Winskel. *The formal semantics of programming languages: an introduction.* MIT press, 1993.

24 Fabio Zanasi. *Interacting Hopf Algebras: the theory of linear systems.* PhD thesis, École Normale Supérieure, 2015.

A Proofs for Section 4

Details for definition 20. By naturality of the symmetries, the vertex generators commute with any permutation matrix P:  .

Thus, we can show that $\Gamma = (k, [A], B, C, D, [E])$ and $\Gamma' = (k, [A], B, CP^T, PD, [PEP^T])$ represent the same open graph.

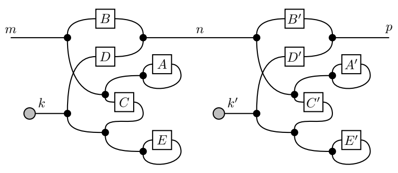


◀

▶ **Lemma 33.** *$\mathcal{A}$ is a prop.*

Proof. We start by proving that $\mathcal{A}$ is a category. The diagram below can be rewritten, using the axioms of $\mathbf{B}$, as a diagram of the form shown in Definition 20. The components of the normal form obtained in this way give the algebraic definition of the composition.

$$\Gamma'\circ\Gamma = \left(k + k', \left[A + BA'B^T\right], BB', (C + B(A' + A'^T)D^T|BC'), \left(\begin{smallmatrix} DB' \\ D' \end{smallmatrix}\right), \left[\left(\begin{smallmatrix} E+DA'D^T & DC' \\ \mathbb{0} & E' \end{smallmatrix}\right)\right]\right)$$



Identities are defined in the obvious way: $\mathbb{1}_n = (0, [\mathbb{0}_n], \mathbb{1}_n, !, ¡, [(\ )])$.

The definition of composition is coherent with the equivalence classes because, whenever $\Gamma \sim \Gamma_0$ with matrix P and $\Gamma' \sim \Gamma'_0$ with matrix P', $\Gamma' \circ \Gamma \sim \Gamma'_0 \circ \Gamma_0$ with matrix $\left(\begin{smallmatrix} P & \mathbb{0} \\ \mathbb{0} & P' \end{smallmatrix}\right)$. Composition is associative because the matrices relative to the vertices are [ ]-equivalent. Clearly, composition is unital and we proved that $\mathcal{A}$ is a category. Now we prove that it is monoidal.

Lead by the interpretation of the matrices that define a morphism, we define monoidal product as follows.

$$\Gamma \otimes \Gamma' = \left(k + k', \left[\left(\begin{smallmatrix} A & \mathbb{0} \\ \mathbb{0} & A' \end{smallmatrix}\right)\right], \left(\begin{smallmatrix} B & \mathbb{0} \\ \mathbb{0} & B' \end{smallmatrix}\right), \left(\begin{smallmatrix} C & \mathbb{0} \\ \mathbb{0} & C' \end{smallmatrix}\right), \left(\begin{smallmatrix} D & \mathbb{0} \\ \mathbb{0} & D' \end{smallmatrix}\right), \left[\left(\begin{smallmatrix} E & \mathbb{0} \\ \mathbb{0} & E' \end{smallmatrix}\right)\right]\right)$$

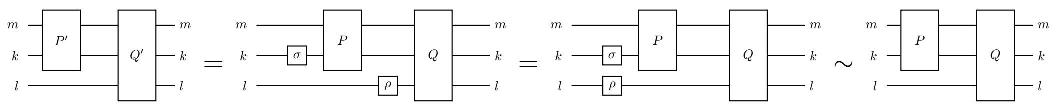
The monoidal unit is the empty diagram: $\mathbb{I} = (0, [(\ )], (\ ), (\ ), (\ ), [(\ )])$

The monoidal product is well-defined on equivalence classes because, whenever $\Gamma \sim \Gamma_0$ with matrix P and $\Gamma' \sim \Gamma'_0$ with matrix P', $\Gamma \otimes \Gamma' \sim \Gamma_0 \otimes \Gamma'_0$ with matrix $\left(\begin{smallmatrix} P & \mathbb{0} \\ \mathbb{0} & P' \end{smallmatrix}\right)$. Clearly, monoidal product is strictly associative and unital. Therefore, the pentagon and the triangle equations [18] hold trivially. The monoidal product is a bifunctor because $(\Gamma_0 \circ \Gamma) \otimes (\Gamma'_0 \circ \Gamma_0) \sim (\Gamma_0 \otimes \Gamma'_0) \circ (\Gamma \otimes \Gamma')$ with permutation matrix $P = \left(\begin{smallmatrix} \mathbb{1} & \mathbb{0} & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & \mathbb{1} & \mathbb{0} \\ \mathbb{0} & \mathbb{1} & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{1} \end{smallmatrix}\right)$.
Thus, $\mathcal{A}$ is a monoidal category. Finally, we prove that it is symmetric. Let $\sigma_{m,n}$ indicate the symmetry: $\sigma_{m,n} = (0, [\mathbb{0}], \left(\begin{smallmatrix} \mathbb{0} & \mathbb{1}_m \\ \mathbb{1}_n & \mathbb{0} \end{smallmatrix}\right), !, ¡, [(\ )])$

Clearly, the symmetry is its own inverse: $\sigma_{m,n} \circ \sigma_{n,m} = \mathbb{1}_{m+n}$.
Moreover, σ is natural as $\sigma_{n,n'} \circ (\Gamma \otimes \Gamma') \sim (\Gamma' \otimes \Gamma) \circ \sigma_{m,m'}$ with permutation matrix $P = \left(\begin{smallmatrix} \mathbb{0} & \mathbb{1} \\ \mathbb{1} & \mathbb{0} \end{smallmatrix}\right)$.
Lastly, the symmetry satisfies the hexagon equations. Thus, $\mathcal{A}$ is a symmetric monoidal category whose objects are natural numbers. In other words, it is a prop. ◀

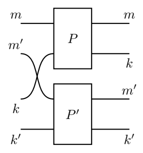
▶ **Lemma 34.** $\mathbf{b}\mathbb{P}$ *is a prop.*

Proof. The proof proceeds exactly as the previous one. We will use diagrammatic calculus of **Mat** for the permutation matrix of the morphisms in order to make the proofs more readable. We start by proving that $\mathbf{b}\mathbb{P}$ is a category. Composition is well-defined on equivalence classes by the monoidal structure of **Mat**. Let $(k, P) \sim (k, \left(\begin{smallmatrix} \mathbb{1}_m & \mathbb{0} \\ \mathbb{0} & \sigma \end{smallmatrix}\right) P) = (k, P')$ and $(l, Q) \sim (l, \left(\begin{smallmatrix} \mathbb{1}_{m+k} & \mathbb{0} \\ \mathbb{0} & \rho \end{smallmatrix}\right) Q) = (l, Q')$.

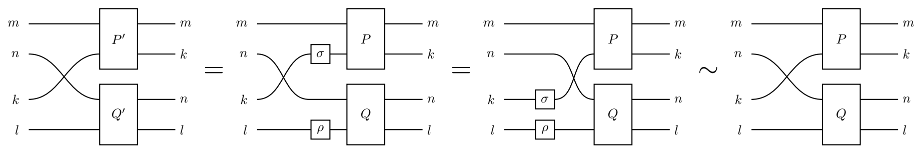


with permutation matrix $\left(\begin{smallmatrix} \mathbb{1}_m & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \sigma & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & \rho \end{smallmatrix}\right)$. Composition is clearly associative and unital because it is associative and unital in **Mat**. The monoidal product is defined with a symmetry on the left because we need to keep track of which of the inputs are bound.

$$(k, P) \otimes (k', P') = \left(k + k', \begin{pmatrix} \mathbb{1}_m & \mathbb{0} & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & \mathbb{1}_{m'} & \mathbb{0} \\ \mathbb{0} & \mathbb{1}_k & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{1}_{k'} \end{pmatrix} \left(\begin{smallmatrix} P & \mathbb{0} \\ \mathbb{0} & P' \end{smallmatrix}\right)\right)$$

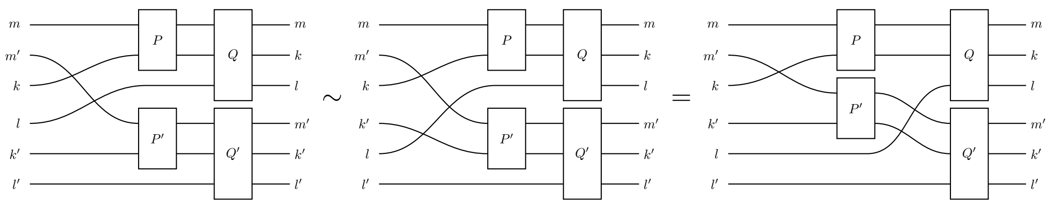


The monoidal unit is the empty diagram: $\mathbb{I} = (0, (\ ))$. The monoidal product is well-defined on equivalence classes by naturality of the symmetries in **Mat**. Let $(k, P) \sim (k, \left(\begin{smallmatrix} \mathbb{1}_m & \mathbb{0} \\ \mathbb{0} & \sigma \end{smallmatrix}\right) P) = (k, P')$ and $(l, Q) \sim (l, \left(\begin{smallmatrix} \mathbb{1}_n & \mathbb{0} \\ \mathbb{0} & \rho \end{smallmatrix}\right) Q) = (l, Q')$.



with permutation matrix $\left(\begin{smallmatrix} \mathbb{1}_{m+n} & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \sigma & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & \rho \end{smallmatrix}\right)$. The monoidal product is a functor because we can change the order in which we enumerate the vertices and because symmetries are natural in

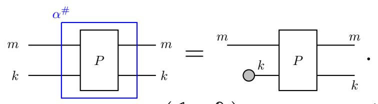
Mat.



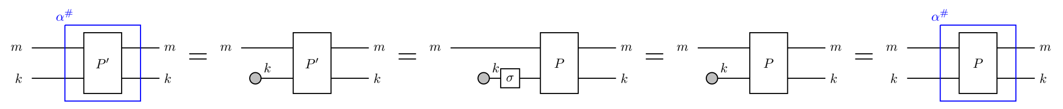
with matrix $\begin{pmatrix} \mathbb{1}_{m+m'+k} & \mathbb{0} & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & \mathbb{1}_{k'} & \mathbb{0} \\ \mathbb{0} & \mathbb{1}_l & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{1}_{l'} \end{pmatrix}$. The monoidal product is clearly unital. The symmetry is lifted from **Mat**: $\sigma_{m,n} = (0, \left(\begin{smallmatrix} \mathbb{0} & \mathbb{1}_m \\ \mathbb{1}_n & \mathbb{0} \end{smallmatrix}\right))$. The symmetry is its own inverse and it satisfies the hexagon equations because it does so in **Mat**.
Therefore, $\mathbf{b}\mathbb{P}$ is a prop. ◀

▶ **Lemma 35.** $\mathbf{b}\mathbb{P}$ *is isomorphic to the free prop on one generator* $0 \to 1$.

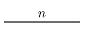
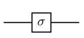
Proof. Define $\alpha^{\#} \colon \mathbf{b}\mathbb{P} \longrightarrow \mathbb{P}$ to be $\alpha^{\#}(k, P) = P \circ (\mathbb{1}_m \otimes \bigotimes_k v)$, where $P \in \mathbb{P}$ is the product of the symmetries that form P in $\mathbf{b}\mathbb{P}$. Diagrammatically,

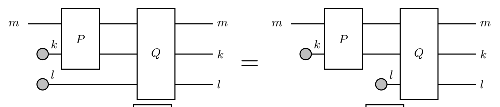


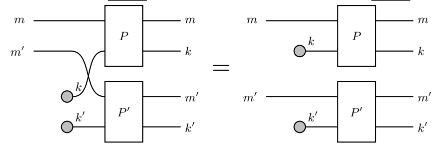
We prove that $\alpha^{\#}$ is well-defined on equivalence classes. Let $(k, P) \sim (k, \left(\begin{smallmatrix} \mathbb{1}_m & \mathbb{0} \\ \mathbb{0} & \sigma \end{smallmatrix}\right) P) = (k, P')$.



We show graphically that $\alpha^{\#}$ is a prop homomorphism

$\alpha^{\#}(\mathbb{0}) = \quad = \mathbb{0}$ $\alpha^{\#}(\mathbb{1}_n) =$  $= \mathbb{1}_n$ $\alpha^{\#}(0, \sigma) =$  $= \sigma$

$\alpha^{\#}((l, Q) \circ (k, P)) =$  $= \alpha^{\#}(l, Q) \circ \alpha^{\#}(k, P)$

$\alpha^{\#}((k, P) \otimes (k', P')) =$  $= \alpha^{\#}(k, P) \otimes \alpha^{\#}(k', P')$

and, by its definition,

$$\alpha^{\#}(\phi) = v$$

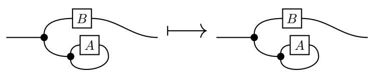
Moreover, $\alpha^{\#}$ is the unique morphism $\mathbf{b}\mathbb{P} \to \mathbb{P}$ with this property. In fact, suppose there is $\beta \colon \mathbf{b}\mathbb{P} \to \mathbb{P}$ such that $\beta(\phi) = v$. Then,

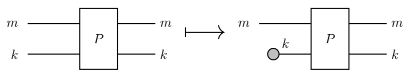
$$\begin{aligned} \beta(k, P) =& \beta((0, P) \circ ((0, \mathbb{1}_n) \otimes (k, \mathbb{1}_k))) = \beta(0, P) \circ (\beta(0, \mathbb{1}_n) \otimes \beta(k, \mathbb{1}_k)) \\ =& P \circ (\mathbb{1}_n \otimes \bigotimes_k v) = a^{\#}(k, P) \end{aligned}$$

Then $\mathbf{b}\mathbb{P}$ is isomorphic to the free prop over one generator $0 \to 1$. ◀

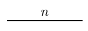
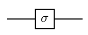
▷ Claim 36. The following are prop homomorphisms.

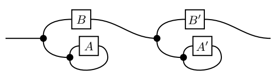
$$i_1 \colon \mathbf{Adj} \longrightarrow \mathcal{A} \qquad\qquad i_2 \colon \mathbf{b}\mathbb{P} \longrightarrow \mathcal{A}$$
$$n \longmapsto n \qquad\qquad n \longmapsto n$$
$$(B, [A]) \longmapsto (0, [A], B, !, ¡, [(\,)]) \qquad\qquad (k, P) \longmapsto (k, [\mathbb{0}_n], P_*^{[1,n]}, \mathbb{0}_{nk}, P_*^{[n+1,n+k]}, [\mathbb{0}_k])$$

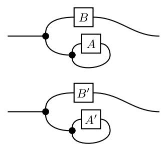


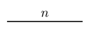
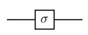


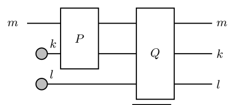
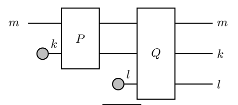
Proof. We prove graphically that they are prop homomorphisms.

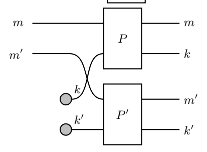
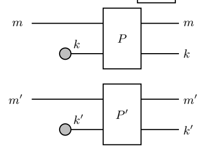
$i_1(\mathbb{I}) = \quad = \mathbb{I}$ $\quad i_1(\mathbb{1}_n) =$  $= \mathbb{1}_n$ $\quad i_1(\sigma, [(\,)]) =$  $= \sigma$

$i_1((B', [A']) \circ (B, [A])) =$  $= i_1(B', [A']) \circ i_1(B, [A])$

$i_1((B, [A]) \otimes (B', [A'])) =$  $= i_1(B, [A]) \otimes i_1(B', [A'])$

$i_2(\mathbb{I}) = \quad = \mathbb{I}$ $\quad i_2(\mathbb{1}_n) =$  $= \mathbb{1}_n$ $\quad i_2(0, \sigma) =$  $= \sigma$

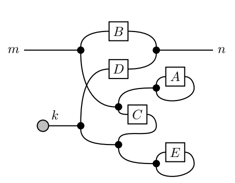
$i_2((l, Q) \circ (k, P)) =$  $=$  $= i_2(l, Q) \circ i_1(k, P)$

$i_2((k, P) \otimes (k', P')) =$  $=$  $= i_2(k, P) \otimes i_2(k', P')$

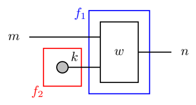
◀

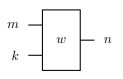
▶ **Lemma 37.** *H, defined on page 12, is a prop homomorphism.*

Proof. Recall that $H \colon \mathcal{A} \to \mathcal{C}$ is identity on objects and $H(k, [A], B, C, D, [E]) = f_1\left(\left(\begin{smallmatrix} B \\ D \end{smallmatrix}\right), \left[\left(\begin{smallmatrix} A & C \\ \mathbb{0} & E \end{smallmatrix}\right)\right]\right) \circ (\mathbb{1}_m \otimes f_2(k, \mathbb{1}_k))$. By calling $w = \left(\left(\begin{smallmatrix} B \\ D \end{smallmatrix}\right), \left[\left(\begin{smallmatrix} A & C \\ \mathbb{0} & E \end{smallmatrix}\right)\right]\right)$, which is a morphism in **Adj**, we can depict the image of H diagrammatically.

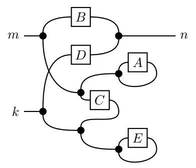


$\longmapsto$

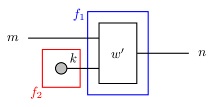
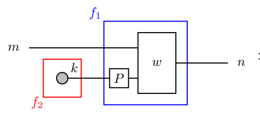
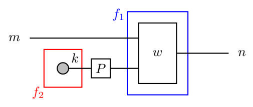


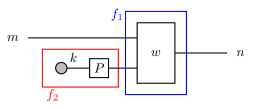
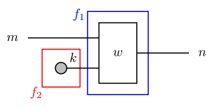


$=$

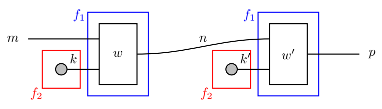


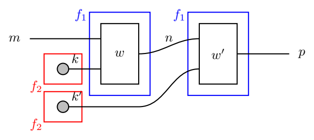
We need to prove that H is well-defined on equivalence classes. Let $\Gamma = (k, [A], B, C, D, [E]) \sim (k, [A], B, CP^T, PD, [PEP^T]) = \Gamma'$.

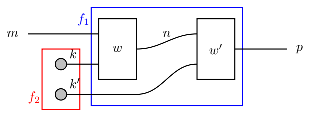
$H(\Gamma') =$  $=$  $=$ 

$=$  $=$  $= H(\Gamma)$

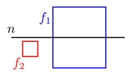
We prove that H is a prop homomorphism. Clearly, H is identity on objects. Moreover, it preserves composition, as it is shown by the diagrams.

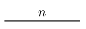
$H(\Gamma') \circ H(\Gamma) =$ 

$=$ 

$=$ 

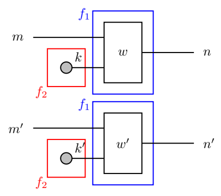
$= H(\Gamma' \circ \Gamma)$

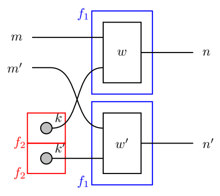
H preserves identities: $H(\mathbb{1}_n) =$ 

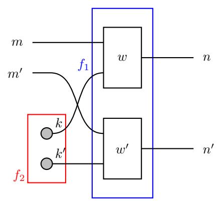
$=$ 

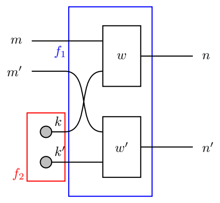
$= \mathbb{1}_n$.

H preserves monoidal product. This is also more clearly seen with string diagrams.

$H(\Gamma) \otimes H(\Gamma') =$ 

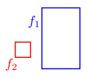
$=$ 

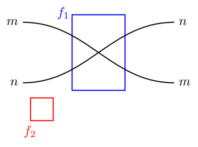
$=$ 

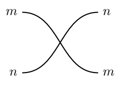
$=$ 

$= H(\Gamma \otimes \Gamma')$

It is easy to show that H preserves monoidal unit and symmetries.

$H(\mathbb{I}) =$ 

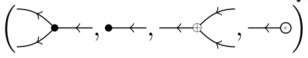
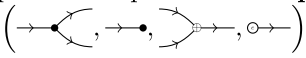
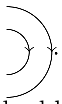
$= \quad = \mathbb{I}$ $\quad H(\sigma_{m,n}) =$ 

$=$ 

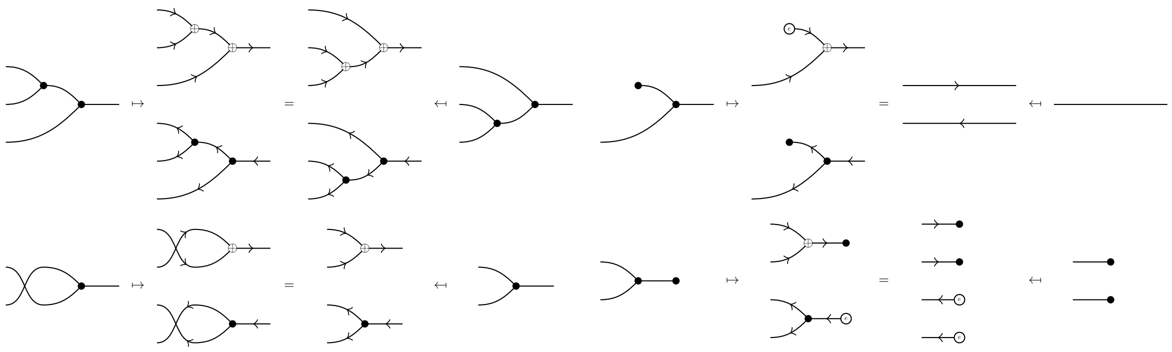
$= \sigma_{m,n}$

◀

B Proofs for Sections 5

Proof of Theorem 27. $F_{\mathcal{N}}$ respects the equations of **Grph** because both the tuples  and  satisfy the commutative bialgebra axioms in figure 2, and they both interact as in figure 3 with the cup .

We explain in detail that the functor $F_{\mathcal{N}}$ preserves associativity of the black monoid, the rest of the equations are written with the same convention. We write on the left-most and right-most sides morphisms in **Grph** that, by associativity, they must be equal. In the centre, we write the morphisms in **Game** to which they are mapped (indicated with $\mapsto$) by the functor $F_{\mathcal{N}}$. These morphisms are equal in **Game** by associativity of the monoid operation $\oplus$ on M and coassociativity of copying. Thus, we can say that $F_{\mathcal{N}}$ preserves associativity of the black monoid.



and similarly for their transposed versions. ◀

Proof of Theorem 29. The proof proceeds by structural induction on Γ.
It is straightforward to check from the definition of $F_{\mathcal{N}}$ that the generators of **Grph** are sent to open games of the required form.
We need to check that composition is of the form as in the statement. We compute explicitly its play, coplay and best response functions.

$$\mathbb{P}_{F_{\mathcal{N}}(\Gamma'\circ\Gamma)}((\underline{\sigma},\underline{\sigma}'),\underline{x}) = \mathbb{P}_{F_{\mathcal{N}}(\Gamma')}(\underline{\sigma}', B^T\underline{x}\oplus D^T f(\underline{\sigma})) = (BB')^T\underline{x}\oplus \left(\begin{smallmatrix} DB' \\ D' \end{smallmatrix}\right)^T f(\left(\begin{smallmatrix} \underline{\sigma} \\ \underline{\sigma}' \end{smallmatrix}\right))$$

$$\begin{aligned}&\mathbb{C}_{F_{\mathcal{N}}(\Gamma'\circ\Gamma)}((\underline{\sigma},\underline{\sigma}'),\underline{x},\underline{q}) = \mathbb{C}_{F_{\mathcal{N}}(\Gamma)}(\underline{\sigma},\underline{x},\mathbb{C}_{F_{\mathcal{N}}(\Gamma')}(\underline{\sigma}',\mathbb{P}_{F_{\mathcal{N}}(\Gamma)}(\underline{\sigma},\underline{x}),\underline{q}))\\ &= ((A+BA'B^T)+(A+BA'B^T)^T)\underline{x}\oplus BB'\underline{q}\oplus(C+B(A'+A'^T)D^T|BC')f(\left(\begin{smallmatrix} \underline{\sigma} \\ \underline{\sigma}' \end{smallmatrix}\right))\end{aligned}$$

$$\begin{aligned}&(\underline{\rho},\underline{\rho}')\in\mathbb{B}_{F_{\mathcal{N}}(\Gamma'\circ\Gamma)}(\underline{x},\kappa)\\ \Leftrightarrow&(\underline{\sigma},\underline{\sigma}')\in\mathbb{B}_{F_{\mathcal{N}}(\Gamma)}(\underline{x},\kappa\circ F_{\mathcal{N}}(\Gamma')_{\underline{\tau}})\wedge(\underline{\tau},\underline{\tau}')\in\mathbb{B}_{F_{\mathcal{N}}(\Gamma')}(F_{\mathcal{N}}(\Gamma)_{\underline{\sigma}}\circ\underline{x},\kappa)\\ \Leftrightarrow&\forall a=1,...,k+k'\\ &\qquad s\in\operatorname*{argmax}_{s\in X} g\Big(s,\left(\begin{smallmatrix} C^T+D(A'+A'^T)B^T \\ C'^TB^T \end{smallmatrix}\right)^a_*\underline{x}\\ &\oplus\left(\begin{smallmatrix} DB' \\ D' \end{smallmatrix}\right)^a_*\kappa((BB')^T\underline{x}\oplus(B'^TD^T|D'^T)f(\underline{\rho}[a\mapsto\rho'_a]))\\ &\quad\oplus\left(\begin{smallmatrix} E+E^T+D(A'+A'^T)D^T & DC' \\ C'^TD^T & E'+E'^T \end{smallmatrix}\right)^a_* f(\underline{\rho}[a\mapsto\rho'_a])\Big)\end{aligned}$$

Similarly, we show that monoidal product has the desired form.

$$\mathbb{P}_{F_{\mathcal{N}}(\Gamma\otimes\Gamma')}((\underline{\sigma},\underline{\sigma}'),(\underline{x},\underline{x}')) = \left(\begin{smallmatrix} B & \mathbb{0} \\ \mathbb{0} & B' \end{smallmatrix}\right)^T \begin{pmatrix} \underline{x} \\ \underline{x}' \end{pmatrix} \oplus \left(\begin{smallmatrix} D & \mathbb{0} \\ \mathbb{0} & D' \end{smallmatrix}\right)^T f(\begin{pmatrix} \underline{\sigma} \\ \underline{\sigma}' \end{pmatrix})$$

$$\begin{aligned} &\mathbb{C}_{F_{\mathcal{N}}(\Gamma\otimes\Gamma')}((\underline{\sigma},\underline{\sigma}'),(\underline{x},\underline{x}'),(\underline{r},\underline{r}')) \\ &= \left(\left(\begin{smallmatrix} A & \mathbb{0} \\ \mathbb{0} & A' \end{smallmatrix}\right) + \left(\begin{smallmatrix} A & \mathbb{0} \\ \mathbb{0} & A' \end{smallmatrix}\right)^T\right) \begin{pmatrix} \underline{x} \\ \underline{x}' \end{pmatrix} \oplus \left(\begin{smallmatrix} B & \mathbb{0} \\ \mathbb{0} & B' \end{smallmatrix}\right) \begin{pmatrix} \underline{r} \\ \underline{r}' \end{pmatrix} \oplus \left(\begin{smallmatrix} B & \mathbb{0} \\ \mathbb{0} & B' \end{smallmatrix}\right) f(\begin{pmatrix} \underline{\sigma} \\ \underline{\sigma}' \end{pmatrix}) \end{aligned}$$

$$\begin{aligned} &(\underline{\rho},\underline{\rho}') \in \mathbb{B}_{F_{\mathcal{N}}(\Gamma'\otimes\Gamma)}((\underline{x},\underline{x}'),\langle\kappa,\kappa'\rangle) \\ \Leftrightarrow &\forall a = \ ...,k+k' \ \rho'_a \in \operatorname*{argmax}_{s\in X} g\Big(s, \left(\left(\begin{smallmatrix} C & \mathbb{0} \\ \mathbb{0} & C' \end{smallmatrix}\right)^T\right)^a_* \begin{pmatrix} \underline{x} \\ \underline{x}' \end{pmatrix} \oplus \left(\left(\begin{smallmatrix} D & \mathbb{0} \\ \mathbb{0} & D' \end{smallmatrix}\right)^T\right)^a_* \langle\kappa,\kappa'\rangle(\left(\begin{smallmatrix} B & \mathbb{0} \\ \mathbb{0} & B' \end{smallmatrix}\right)^T \begin{pmatrix} \underline{x} \\ \underline{x}' \end{pmatrix} \\ &\oplus \left(\begin{smallmatrix} D & \mathbb{0} \\ \mathbb{0} & D' \end{smallmatrix}\right)^T f(\underline{\rho}[a\mapsto s])) \oplus \left(\left(\begin{smallmatrix} E & \mathbb{0} \\ \mathbb{0} & E' \end{smallmatrix}\right) + \left(\begin{smallmatrix} E & \mathbb{0} \\ \mathbb{0} & E' \end{smallmatrix}\right)^T\right)^a_* f(\underline{\rho}[a\mapsto s])\Big) \end{aligned}$$

◀