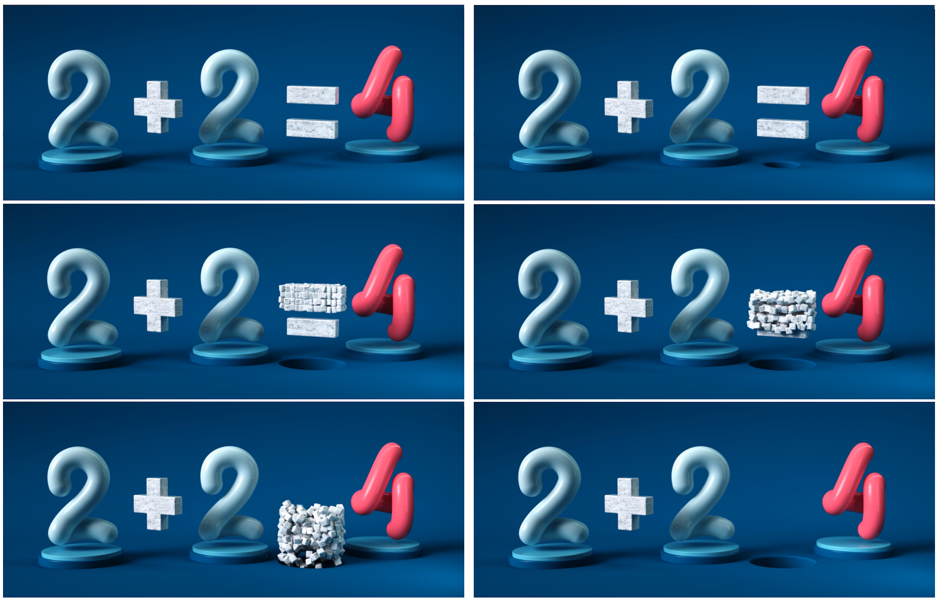
Towards a Generator of Discrete Models in the Form of Flow Networks

Guillaume Paya-Monet



Presented to the Department of Mathematics and to the Department of Computer Science in partial fulfilment of the requirements for a BSc. degree with Honours

École polytechnique

Palaiseau, France

12 April 2020

Table of Contents

List of Figures

Abstract

Weighted directed graphs are useful models for several types of applications involving material or digital elements flowing through an interconnected network. What settings will maximize the amount of elements flowing from source to destination? Complex interactions involving the flow at junctions (or nodes of the network graph) made network flow problems difficult to solve for 50 years.

To our knowledge, no attempt had yet been made to leverage the framework of category theory to model such networks. The benefit would be the unification, clarification and efficiency in the mathematics analyzing such systems, so that we are able to extract the structure of flow networks. This could serve as the foundation of the automatic generation of optimised algorithms traversing such networks. To do so, firstly, we set out to provide an understanding of graphs and their categorisation. This allows to provide the category of a network which we will be using. This would be the initial building block of the presentation of the category of flow networks, as each category form a different network. Secondly, with the formed monoidal category, an equivalence relation is needed to define the relationship between different networks.

The algebraic approach to graph rewriting is based upon category theory. Therefore, the result of this research has the potential to be the enabler of the build and run of a generator of strings that would be specifying any flow network while minimising the complexity of the systems required to deliver such a solution.

Acknowledgements

I would like to express my sincere appreciation and gratitude to my supervisor, Professor Samuel Mimram, for having introduced category theory to me and for having selected such an interesting thesis problem and supervised it so that I could solve it in the allocated time frame, and made sure that I could write it up properly. During this relentless non-linear process, firstly, I learned how to work at the very heart of mathematics and do it on a real-world application, the legendary problem of the maximisation of flow networks. Secondly, I was able to quench my thirst for patterns, structure, abstraction, models, and rigour, in building, at breakneck speed, some very vertical mathematics, and in a new and innovative way. Thirdly, all this has given me the opportunity to reflect on how to define mathematics, applied mathematics and computer science with respect to one another, with, at their centre, models of systems abstracted from real-world objects. Last, my supervisor's deep knowledge of category theory and experience in research at a high level has enabled me to complete this research successfully. Without his support and guidance, even amidst the abrupt reorganisation of his laboratory due to the pandemic lockdown, this thesis, requiring careful handling, would not have been possible.

I am also grateful to the mathematicians and scientists at École polytechnique's LIX Laboratory who accepted to patiently provide multiple feedbacks, insightful comments, helpful ideas and information, and practical advice. Thank you Emmanuel Haucourt, especially, for providing stimulating discussions and helpful ideas, including the much-needed cup of coffee. Thank you also to my professional colleagues and friends: Mihai, François-Xavier, Adrien, Dominique, Diane, Marc, you have helped me tremendously in my research and writing of this thesis. Mathematics once again is such a team sport!

I would like to thank the École polytechnique's Bachelor Team, Claire Lenz, Céline Chalard, for having put together such an excellent academic programme, and for their unrelenting encouragements and assistance. During the open day three years ago, you said to my parents and me "Your kid will be doing research, even while an undergraduate, with top academics in the field." This was true, even beyond my wildest expectations. Now, I know first-hand how important the creation and synthesis of many areas altogether is.

Last but not least, to my parents and Carolina, thank you for your continuous and discreet support along the way.

1 Introduction

Optimising the traversing of a flow of material, and now immaterial or digital elements, through a network from a source to a target has been extensively researched in the second part of 20^{th} century. Flow networks are directed graphs where each edge has a capacity assigned to it. They are used to represent programs where finite data is transferred between two points, whether it be to model the dispatch of merchandise within a railroad network, circulation in water pipes, traffic in computer networks, or trust in social networks. The maximum flow that can pass through such a network can be measured, and many algorithms compute it.

One of the first algorithm to find the maximum flow is the Ford–Fulkerson method (Ford–Fulkerson algorithm or FFA) and it guarantees to find the maximum value. In this thesis we will aim to categorize the mathematical object of flow networks.

1.1 Specific Problem

Unfortunately, flow networks can become too complex to measure the total data that flows in it. When composing a network, usually what is important is the output of the system, the evaluation. In network problems, one is always searching for the computation of an output with regards to a given system. However, when the network becomes too convoluted then the evaluation can be difficult to measure. There are numerous algorithms that are used to compute the maximum flow or any other properties of a network but, as with any algorithms, all of them fall short when we are dealing with large numbers. So a deep understanding of the system is needed before any improvements can be made. Which is where category theory comes in, to have a deeper comprehension of the components of flow networks, which could in turn, in further research, could be applied to the works of flow maximisation.

1.2 Outline

The methodology that we propose to use to solve such challenges is the category theory framework. Category theory is a relatively new adjunction of the field of mathematics that is explained by its evolution since the beginning of the 19th century. Category theory was founded in the mid 1940's by S. MC LANE and S. EILENBERG, with the lofty goals of unification, clarification and efficiency in mathematics, then leveraging their immense predictive power.

So, using this approach, we begin our thesis by setting out to provide an application of category theory to graphs. But first, an understanding of the components of graphs and the theory of category theory is required as the category of network will be comprised using it. At this stage, only simple networks are defined which leaves out the most important properties of flow networks, the capacitance. This is why an additional category of weighted networks will be built upon the category of simple networks. This allows to provide the category of network which we will be using for the category of flow networks. Following our category of weighted networks, we need to understand what flow means in this scenario. An additional category, a category of observations of flow, will then be constructed in parallel. Thanks to the category of weighted networks and observations we are now able to attach the concept of flow to a network with capacitance. Thus, completing flow networks. This would be the initial building block of the presentation of the category of flow networks, as each category form a different network and thanks to the generators and their relations provided by the presentation, we would be able to construct any flow network needed.

1.3 Summary

Finally, after all the steps taken to discuss in great detail the category of flow networks and the equivalence of flow networks, a presentation of the category of flow networks could be established, which would be completed by the generators and the relations between the generators to provide for the composition any flow network. Indeed, the algebraic approach to automated graph rewriting is based upon category theory. Hence, the theory made in this thesis, now coupled with rewriting theory, could potentially simplify the work needed to compute the "answer" to the problem, in our case that would be finding the maximum flow.

It is though worth remembering that the evaluation of a network is not the only important dimension to monitor. Leveraging the complexity framework, the present theory of flow networks can help simplify complex systems and determine its complexity, keeping in mind that the complexity of the diagrams, and more specifically the complexity of the circuits, is one of the key points to address. [BI16]

As an overall conclusion, and perhaps most importantly, with the result of this research, we will lead the path to enable the build and run of a generator of flow networks; this object specifying any optimized flow network, while minimizing the complexity of the systems required to deliver such a solution.

2 Flow Networks

2.1 Statements of Problem

Definition 1. As defined by L. R. Ford Jr. [For56], a network flow is a collection of nodes P_i for $i \in [| \, 0, N \, |]$, some of which may be joined by an edge $e_{ij} \in E$ joining P_i to P_j where E is the set of all edges. Associated to each arc is a capacity c_{ij} and a length l_{ij} which could, according to different situations, represent the distance or the time required to go from P_i to P_j, but the length will not be used in this thesis. Networks described by Ford also have an origin P_0 and a terminal P_N. The flow from P_i to P_j is denoted as $x_{ij} \geq 0$. Furthermore, x_F is the maximum flow that passes through P_0 and his work consisted of maximizing this.

$$\sum_j x_{0j} - \sum_j x_{j0} = x_F$$

2.2 Properties

However, some properties relating to flow networks are needed for the following work to function as what was found in this bachelor thesis pertains to networks that are in accordance to flow network rules.

In a flow network, for every node P_j, the flow that goes into it is equal to the flow out of that node:

$$\sum_{i|e_{ij}\in E} x_{ij} = \sum_{w|e_{jw}\in E} x_{jw}$$

We also have that the flow running through an edge cannot exceed the capacity of that edge:

$$\forall i,j \in [|\ 0,N\ |] \mid e_{ij} \in E, c_{ij} \leq x_{ij}$$

3 The Category of Flow Networks

Now that we have established a basis of what a flow network is, we can now apply the tools of category theory to abstract its mathematical structure.

Therefore, in **this Chapter**, in order to establish the category of flow networks, we will need to look at the category of graphs in general and see how it can be applied to networks.

3.1 Definitions

It is important, however, to look at the basic definitions of category theory, from categories to natural transformations. Definitions in **this Chapter** get some of their inspiration from Tom Leinster [Lei14] and Saunders Mac Lane [ML71].

3.1.1 Categories

Definition 2. A **category** $\mathcal{C}$ consists of:

- A collection of **objects** which are written as $ob(\mathcal{C})$
- For each pair $A, B \in ob(\mathcal{C})$, a collection $\mathcal{C}(A,B)$ of morphisms from A to B.
- We now need to define composition such that for each $A, B, C \in ob(\mathcal{C})$, the composition law is

$$\begin{aligned}\mathcal{C}(B,C)\times\mathcal{C}(A,B) &\to \mathcal{C}(A,C)\\ (g,f) &\mapsto g\circ f\end{aligned}$$

- For each $A \in ob(\mathcal{C})$, there exists an element $1_A \in \mathcal{C}(A,A)$ called the identity.
- We need the composition to be associative, meaning that for $f \in \mathcal{C}(A,B)$, $g \in \mathcal{C}(B,C)$ and $h \in \mathcal{C}(C,D)$

$$(h\circ g)\circ f = h\circ(g\circ f)$$

- The identity has to be neutral, meaning that for each $f \in \mathcal{C}(A,B)$, we have that

$$f\circ 1_A = f = 1_B \circ f$$

3.1.2 Functors

When given mathematical objects, it is always sensible to ask how to map between said objects. Here, functors go from a category to another. To define what a functor $\boldsymbol{F}$ is, we will need to instantiate two categories $\mathcal{C}$ and $\mathcal{D}$ such that $\boldsymbol{F} : \mathcal{C} \to \mathcal{D}$.

Definition 3. This map consists of:

- A function between the objects of the categories $ob(\mathcal{C}) \to ob(\mathcal{D})$, where for an item $C \in \mathcal{C}$ we have $C \mapsto \boldsymbol{F}(C)$.
- Two elements $C, C' \in \mathcal{C}$, a map $\mathcal{C}(C, C') \to \mathcal{D}(\boldsymbol{F}(C), \boldsymbol{F}(C'))$ written as $f \mapsto \boldsymbol{F}(f)$.

We then need the following axioms to be satisfied:

- $\boldsymbol{F}(f \circ f') = \boldsymbol{F}(f) \circ \boldsymbol{F}(f')$ whenever we have $C \xrightarrow{f'} C' \xrightarrow{f} C''$ for $C, C', C'' \in ob(\mathcal{C})$
- For $C \in ob(\mathcal{C})$, we have $\boldsymbol{F}(1_C) = 1_{\boldsymbol{F}(C)}$.

3.1.3 Natural Transformations

We know that functors represent maps between categories but there exists a notion of maps between functors, called natural transformations. Let two categories $\mathcal{A}$ and $\mathcal{B}$, such that there exists two functors F and G between the two categories (as shown in figure 1).

Definition 4. A **natural transformation** $\alpha : F \to G$ is a family $F(A) \xrightarrow{\alpha_A} G(A))_{A \in \mathcal{A}}$ of maps $\mathcal{B}$ such that for every map $A \xrightarrow{f} A'$ in $\mathcal{A}$, the square (shown in figure 2) commutes.

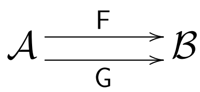


Figure 1: Functor relations between two categories

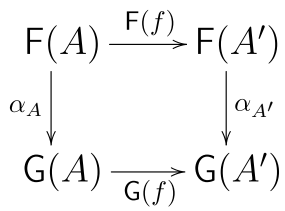


Figure 2: Natural transformation commutative property

Having established the category theory theoretical definitions, we are now ready to progress to the thesis, starting with the category of directed graphs

3.2 Category of Graphs

Before moving to the category of networks, we have to look at the category of directed graphs. We know that a directed graph has vertices and edges that go from one vertex to another but not necessarily in reverse.

Definition 5. Using as a basis B. Bollobas [Bol98] and L. H. Harper [Har80b], we define a directed graph to have the following properties:

- A set of vertices V.
- A set of edges E;
- A function δ_+ that maps an edge to the vertex which is at the head of the edge.
- A function δ_- that maps an edge to the vertex which is at the tail of the edge.

To complete the category of directed graphs, we need to introduce a second category, the category **Set**.

Definition 6. Using as support the work from Mac Lane [ML71], we can define the Category **Set** as follows:
The category **Set** denotes the category whose objects are all sets, the morphisms between two sets are the set of functions and the composition is the regular composition of functions.

Therefore, the Category of directed graphs or **Digraphs** is the functor category whose domain is the diagram category and the codomain is the **Set** category (as shown in figure 4)

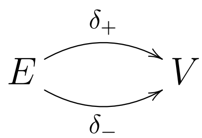


Figure 3: Diagram Category

$$Funct(E \rightrightarrows V, Set)$$

Figure 4: Category of directed graphs

3.3 Proof of Digraphs

Proof. To prove the Category of directed graphs, let us denote $\mathcal{C}$ the category of diagrams with objects E and V and morphisms δ_+ and δ_-. Then denote the functor from $\mathcal{C}$ to **Set** as $F : \mathcal{C} \to \textbf{Set}$.

This means that we have:

- $F(E)$ which is the set of edges in a new graph.
- $F(V)$ which is the set of vertices in a new graph.
- $F(\delta_+)(e)$ which represents the function that maps an edge $e \in E$ to its head vertex $v \in V$ in a new graph.

- $F(\delta_-)(e)$ which represents the function that maps an edge $e \in E$ to its tail vertex $v \in V$ in a new graph.

To complete the Category of graphs **Digraphs** we need morphisms which are going to be natural transformations. Denote two graphs G_1 and G_2 with the following functions between sets of edges and functions between sets of vertices.

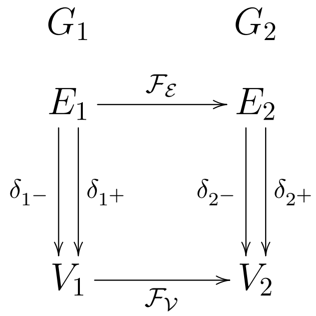


Figure 5: Functor relations between two graphs

Consider the construction of the graphs. To go from one graph to another, we need the functor of the set of vertices of the head of the edge for the first graph to be equal to the head of the functor of the set of edges of an edge for the second graph, similarly to the tail of an edge (as shown in figure 6).

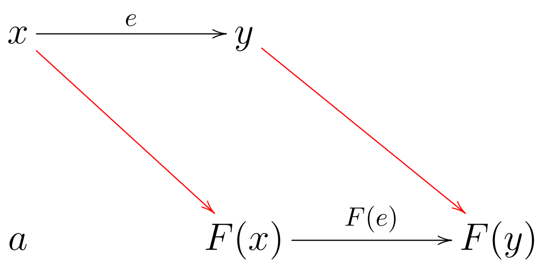


Figure 6: Functor relations between two graphs

Meaning that

$$f_V(\delta_{1+}(e)) = \delta_{2+}(f_E(e))$$
$$f_V(\delta_{1-}(e)) = \delta_{2-}(f_E(e))$$

Which proves the commutative law needed for natural transformations.

□

3.4 Properties

Now we have to expand on the **category of digraphs** to reach the stage where we build the **category of networks**. The particularity of a network, that was not defined in the category of graph, but is needed for the composition of networks, is the fact that they have entry and exit vertices. Therefore we need to incorporate these properties and define the category of networks in terms of the sources and sinks (as show in figure 7).

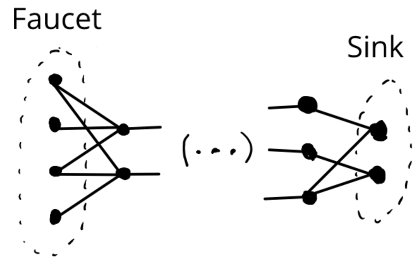


Figure 7: Example of a network

3.5 The Category of Networks

Definition 7. **Network theory** is the study of graphs as a representation of either symmetric relations (undirected graphs) or asymmetric relations (directed graphs) between discrete objects.

In computer science and network science, **network theory** is a part of **graph theory**: a **network** can be defined as a graph in which nodes and/or edges have attributes (e.g. names). It could be also regarded as a **weighted directed graph**.

Definition 8. For the purpose of our research here, we call a network to be a **simple network** if it is a directed graph, or a **weighted network** if it is a weighted directed graph.

In **this Chapter**, we discuss **flow networks** (or **transportation networks**) by imposing additional properties to our definition of a weighted network.

Definition 9. From here we can then construct the **category $\mathcal{N}$ of simple networks** with the following properties:

- $ob(\mathcal{N}) = \mathbb{N}$
- $(\forall)\, m, n \in ob(\mathcal{N})$, we define the morphisms $\mathcal{N}(m, n)$ the set of graphs (V, E, s, t, S, T) with m "entries" and n "exits" such that:
  - V is the set of vertices of the network.
  - E is the set of edges of the network.
  - A function s that maps an edge to the vertex which is at the head of the edge.
  - A function t that maps an edge to the vertex which is at the tail of the edge.
  - S is a subset of V, called **entry vertices**, endowed with a specified total order such that $|S| = m$.

– T is a subset of V, called **exit vertices**, endowed with a specified total order such that $|T| = n$.

– S and T are disjoint non-empty subsets of V.

Observation 10. We need to have a total order for the set of vertices in **Definition 9** as otherwise the composition could change the structure of the graph if there is no clear order of the vertices. As a result, the composition between two graphs will always be the same. As shown in figure 8, if we did not have a total order for the sets of sources and sinks couple with a bijective function, then a different network could result when composing the same two networks.

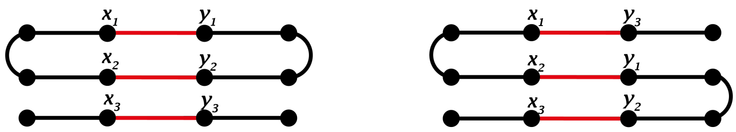


Figure 8: Example of why a total order is needed

To complete the category of networks we need to define composition of the morphisms of the category and the identity.

Definition 11. We then assign the following **composition** to our **category** $\mathcal{N}$, let $m, n, o \in ob(\mathcal{N})$. Now let

- $(V_1, E_1, s, t, S_1, T_1) \in \mathcal{N}(m, n)$
- $(V_2, E_2, s, t, S_2, T_2) \in \mathcal{N}(n, o)$

Therefore

$$\begin{gathered} \mathcal{C}(n,o) \times \mathcal{C}(m,n) \to \mathcal{C}(m,o) \\ ((V_2, E_2, s, t, S_2, T_2), (V_1, E_1, s, t, S_1, T_1)) \mapsto (V_3, E_3, s, t, S_3, T_3) \end{gathered} \tag{1}$$

Such that $V_3 = (V_1 \sqcup V_2) \setminus \{(x, 1) \mid x \in T_1\}$ where the disjoint union is associative up to isomorphisms therefore for a set A, B and C, we have that $A \sqcup (B \sqcup C) = (A \sqcup B) \sqcup C$.

Observation 12. The reason why we require a disjoint union and not a union is because if we use two networks with some vertices that equal, the union would unify the two sets and information would be lost. This is due to the fact that there cannot be any duplicates in sets. With the disjoint union, then all of the vertices are preserved through union.

Let $f : T_1 \to S_2$ a bijective function that maps the i^{th} element of T_1 to the i^{th} element of S_2. Therefore we have that $E_3 = (E_1 \cup E_2)$ where ($\forall e \in E_1$ such that $t(e) \in T_1$, then $t(e) = f(t(e))$). We also have that $S_3 = S_1$ and $T_3 = T_2$.

This means that all the exit vertices of a first graph are replaced by the entry vertices of the second graph and the edges of the first graph are reorganised accordingly (as shown in figure 9).

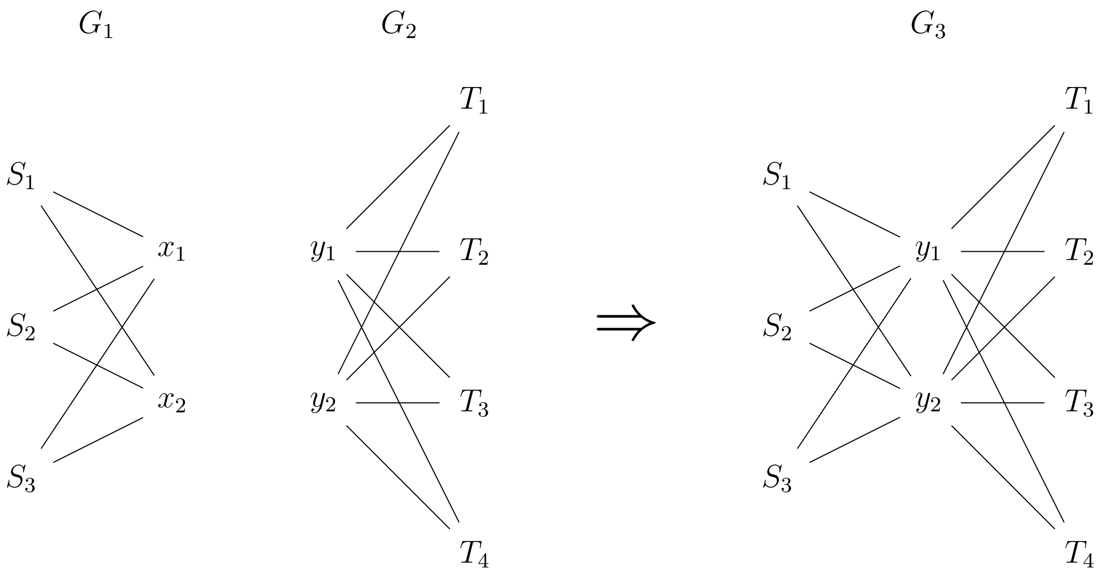


Figure 9: Example of composition of networks

Definition 13. To complete the **category** $\mathcal{N}$, the **identity** of morphisms, will be formed as followed: Let $m \in ob(\mathcal{N})$, the identity is then

$$(V, \emptyset, s, t, S, S) \in \mathcal{N}(m, m) \quad ,$$

where $S = T = V$ (as shown in figure 10).

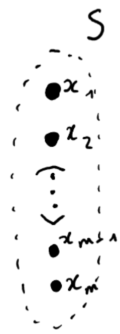


Figure 10: Network identity

This means that for the identity we identify the entry vertices, S, with the exit vertices, T, because there are no edges between S and T. Both S and T are the size of V, because the identity has no other vertices.

3.6 Proof of the Category of Simple Networks

To prove that what we constructed is indeed a category, we need to prove the associativity and identity of the composition.

Proof of the Composition

Proof. Let us first show that the **composition** is indeed associative: Take $m, n, o, p \in ob(\mathcal{N})$ such that $(V_1, E_1, s, t, S_1, T_1) \in \mathcal{N}(m, n)$, $(V_2, E_2, s, t, S_2, T_2) \in \mathcal{N}(n, o)$ and $(V_3, E_3, s, t, S_3, T_3) \in \mathcal{N}(o, p)$.

Therefore we want to show that

$$((V_3, E_3, s, t, S_3, T_3) \circ (V_2, E_2, s, t, S_2, T_2)) \circ (V_1, E_1, s, t, S_1, T_1) = \\ (V_3, E_3, s, t, S_3, T_3) \circ ((V_2, E_2, s, t, S_2, T_2) \circ (V_1, E_1, s, t, S_1, T_1))$$

First left us compute the left hand side of the equality:

$$\begin{aligned}
&((V_3, E_3, s, t, S_3, T_3) \circ (V_2, E_2, s, t, S_2, T_2)) \circ (V_1, E_1, s, t, S_1, T_1) = \\
&((V_2 \sqcup V_3) \setminus \{(x,1) \mid x \in T_2\}, (E_2 \cup E_3), s, t, S_2, T_3) \circ (V_1, E_1, s, t, S_1, T_1) = && (!) \\
&(V_1 \sqcup ((V_2 \sqcup V_3) \setminus \{(x,1) \mid x \in T_2\}) \setminus \{(x,1) \mid x \in T_1\}, E_1 \cup E_2 \cup E_3, s, t, S_1, T_3) = && (!!) \\
&(((V_1 \sqcup (V_2 \sqcup V_3)) \setminus \{(x,1) \mid x \in T_2\}) \setminus \{(x,1) \mid x \in T_1\}, E_1 \cup E_2 \cup E_3, s, t, S_1, T_3) = \\
&((V_1 \sqcup (V_2 \sqcup V_3)) \setminus (\{(x,1) \mid x \in T_2\} \cup \{(x,1) \mid x \in T_1\}), E_1 \cup E_2 \cup E_3, s, t, S_1, T_3)
\end{aligned}$$

(!) Such that $(\forall e \in E_2 \mid t(e) \in T_2 \text{ then } t(e) = f(t(e)))$
(!!) Such that $(\forall e \in E_1 \mid t(e) \in T_1 \text{ then } t(e) = f(t(e)))$

Now let us compute the right hand side of the equality:

$$\begin{aligned}
&(V_3, E_3, s, t, S_3, T_3) \circ ((V_2, E_2, s, t, S_2, T_2) \circ (V_1, E_1, s, t, S_1, T_1)) = \\
&(V_3, E_3, s, t, S_3, T_3) \circ ((V_1 \sqcup V_2) \setminus \{(x,1) \mid x \in T_1\}, (E_1 \cup E_2), s, t, S_1, T_2) = && (!) \\
&((((V_1 \sqcup V_2) \setminus \{(x,1) \mid x \in T_1\}) \sqcup V_3) \setminus \{(x,1) \mid x \in T_2\}, E_1 \cup E_2 \cup E_3, s, t, S_1, T_2) = && (!!) \\
&((((V_1 \sqcup V_2) \sqcup V_3) \setminus \{(x,1) \mid x \in T_1\}) \setminus \{(x,1) \mid x \in T_2\}, E_1 \cup E_2 \cup E_3, s, t, S_1, T_2) = \\
&((((V_1 \sqcup V_2) \sqcup V_3) \setminus (\{(x,1) \mid x \in T_1\}) \cup \{(x,1) \mid x \in T_2\}), E_1 \cup E_2 \cup E_3, s, t, S_1, T_2)
\end{aligned}$$

(!) Such that $(\forall e \in E_1 \mid t(e) \in T_1 \text{ then } t(e) = f(t(e)))$
(!!) Such that $(\forall e \in E_1 \cup E_2 \mid t(e) \in T_2 \text{ then } t(e) = f(t(e)))$

Since we have assumed that the disjoint union was associative up to isomorphisms, then

$$((V_1 \sqcup (V_2 \sqcup V_3)) \setminus (\{(x,1) \mid x \in T_2\} \cup \{(x,1) \mid x \in T_1\}), E_1 \cup E_2 \cup E_3, s, t, S_1, T_3) = \\ ((((V_1 \sqcup V_2) \sqcup V_3) \setminus (\{(x,1) \mid x \in T_1\}) \cup \{(x,1) \mid x \in T_2\}), E_1 \cup E_2 \cup E_3, s, t, S_1, T_2)$$

Therefore the associativity of the composition has been proven. □

Proof of the Identity Morphism

Proof. It is now required to prove that the identity is neutral: Take $m, n \in ob(\mathcal{N})$ such that $(V_1, E_1, s, t, S_1, T_1) \in \mathcal{N}(m, n)$, $1_m = (V_2, \emptyset, s, t, S_2, S_2) \in \mathcal{N}(m, m)$ and $1_n = (V_3, \emptyset, s, t, S_3, S_3) \in \mathcal{N}(n, n)$. We will consider in this case that graphs are equivalent up to an isomorphism meaning that two graphs of the same structure are equal.

In the first case:

$$\begin{aligned}(V_1, E_1, s, t, S_1, T_1) \circ 1_m &= ((V_2 \sqcup V_1) \setminus \{(x, 1) \mid x \in S_2\}, \emptyset \cup E_1, s, t, S_1, T_1) \\ &= (V_1, E_1, s, t, S_1, T_1)\end{aligned}$$

In the second case:

$$\begin{aligned}1_n \circ (V_1, E_1, s, t, S_1, T_1) &= ((V_1 \sqcup V_3) \setminus \{(x, 1) \mid x \in T_1\}, E_1 \cup \emptyset, s, t, S_1, S_3) \quad (!) \\ &= (V_1, E_1, s, t, S_1, T_1)\end{aligned}$$

(!) Such that $(\forall e \in E_1 \mid t(e) \in T_1 \text{ then } t(e) = f(t(e)))$

Therefore the identity is neutral. □

As associativity and neutrality of the identity have been shown, then the **category of simple networks** has been proven.

3.7 Expanding the Category of Simple Networks to Weighted Networks

The difference between a simple network and a flow network[1] we use in **this Chapter** is the fact that flow networks hold additional properties, such as each edge receives a flow and each edge holds a maximum capacity (weight) that the flow cannot exceed.

Therefore the category that we have found for simple networks has to be adjusted so that we can accommodate a capacity function.

We introduce the category of weighted networks in this section, then we discuss the flows in weighted networks later in **this Chapter** when we will introduce the **category of observations**, and we link these two categories **this Chapter** to characterise flow networks.

I had to characterise what equivalency between flow networks meant. To this effect, I tried different direct approaches that didn't provide any positive results.

After exploring the situation with my supervisor, Samuel Mimram, we agreed to use a category of observations to introduce progressively the notion of flow.

In practice, the only parameters guaranteed to be observable are the flows in and out of a network. By using categories of observations, we take the novel approach to look at flow

[1] also known as a transportation network

networks only by their observable parameters. By posing this restriction, we do not limit the usability of the results but make the proof tractable.

We will then define the **category of weighted networks** $\mathcal{W}$ by extending what we have found for the **category of simple networks** $\mathcal{N}$:

Definition 14. Our **category of weighted networks** $\mathcal{W}$ is then described as followed:

- $ob(\mathcal{W}) = \mathbb{N}$
- $(\forall)\, m, n \in ob(\mathcal{W})$, we define the morphisms $\mathcal{W}(m, n)$ the set of graphs

$$\mathcal{W}(m, n) = \{(V, E, s, t, S, T, \mathbf{c}) \mid m, n \in ob(\mathcal{W})\}$$

  with m "entries" and n "exits" such that:

  – V is the set of vertices of the network.

  – E is the set of edges of the network.

  – A function s that maps an edge to the vertex which is at the head of the edge.

  – A function t that maps an edge to the vertex which is at the tail of the edge.

  – S is a subset of V, called **entry vertices**, endowed with a specified total order[2] such that $|S| = m$.

  – T is a subset of V, called **exit vertices**, endowed with a specified total order[3] such that $|T| = n$.

  – S and T are disjoint non-empty subsets of V.

  – $\mathbf{c} : E \to \mathbb{R}$ is a function that maps an edge to its corresponding weight.

Proof. The proof of the category of weighted networks works in the same way as the proof of the category of simple networks as the capacity function does not change the line of argument of the proof. □

4 Equivalence of Flow Networks

4.1 Purpose of Study

Now that the category of weighted networks has been defined, we can study flow networks and continue with the categorisation. We can make the observations on any graph depending on the flow that passes on either side of the network.

Definition 15. A **flow network** (also known as a **transportation network**) is a weighted network $\mathcal{W}$ with morphisms $\mathcal{W}(m, n)$ for $m, n \in \mathcal{W}$, where each edge $e \in E$ has a capacity defined by the function $\mathbf{c} : E \to \mathbb{R}$ and each edge receives a flow. The amount of flow on an edge cannot exceed the capacity of the edge.

[2] see **Observation 10**
[3] see **Observation 10**

Regarding the definition of what we understand by the concept of Ford-Fulkerson [FF56] in the above definition, which is a function in the Ford-Fulkerson paper from the set of arcs (edges), to $\mathbb{R}^+$ subject to flow constraints (flow in = flow out, and not exceeding capacity values), we are going to relax the definition of a flow for now.

We introduce the concept of **observations**, where an observation is a set

$$\{\{x_1, x_2, \dots, x_m\}, \{y_1, y_2, \dots, y_n\}\} \subseteq \mathbb{R}^m \times \mathbb{R}^n$$

assuming that we have m "entry" vertices and n "exit" vertices in the network, and where x_i is the flow in the vertex i for $1 \leq i \leq m$ and y_j is the flow out in the vertex j with $1 \leq j \leq n$.

With this observation, we categorise flow networks as functors from the weighted network category to the observation category, and leave the meaning from **Definition 1**; a flow network is seen here as a functor between categories, rather than simply a weighted network for our purpose.

Let us introduce the **category of observations** in the next section.

4.2 The Category of Observations

Definition 16. The **category of observations Obs** has the following sets of objects and morphisms:

- $ob(\mathbf{Obs}) = \mathbb{N}$ (natural numbers)
- $(\forall)\, m, n \in ob(\mathbf{Obs})$, we define the set of morphisms $\mathbf{Obs}(m, n)$ to be the subspace

  $$\mathcal{S} \subseteq \mathbb{R}^m \times \mathbb{R}^n \quad , \quad \mathcal{S} = \{\{\{x_1, \dots, x_m\}, \{y_1, \dots, y_n\}\} \mid m, n \in ob(\mathbf{Obs})\}$$

  In other words, $f \in \mathbf{Obs}(m, n)$ if for any sets of real numbers $\{x_i\}_i$ and $\{y_j\}_j$, indexed by $\{1, 2, \dots, m\}$ and $\{1, 2, \dots, n\}$, respectively, we have

  $$\{\{x_1, \dots, x_m\}, \{y_1, \dots, y_n\}\} \in f$$

Definition 17. Composition of Morphisms. Let $m, n, p \in ob(\mathbf{Obs})$. Let $f \in \mathbf{Obs}(m, n)$ and $g \in \mathbf{Obs}(n, p)$ be morphisms.

The **composition** $g \circ f$ is the set of

$$\{\{x_1 \dots x_m\}, \{z_1 \dots z_p\}\} \in \mathbb{R}^m \times \mathbb{R}^p$$

if there exists a set $\{y_1 \dots y_n\} \in \mathbb{R}^n$ such that

$$\{\{x_1 \dots x_m\}, \{y_1 \dots y_n\}\} \in f \quad \text{and} \quad \{\{y_1 \dots y_n\}, \{z_1 \dots z_p\}\} \in g$$

We write this formally as:

$$\mathbf{Obs}(n, p) \times \mathbf{Obs}(m, n) = \mathbf{Obs}(m, p) \tag{2}$$

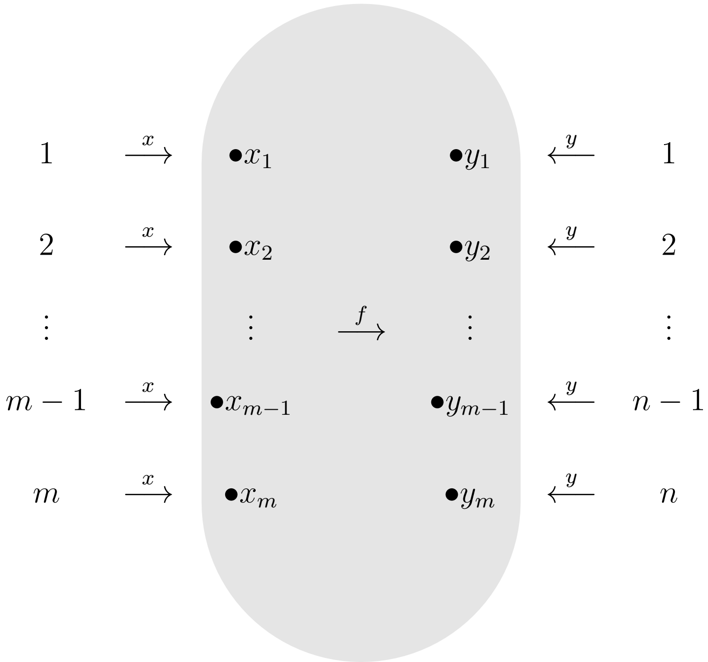


Figure 11: Sketching the morphism f

Definition 18. The Identity Morphism. For each $m \in ob(\mathbf{Obs})$ we have an **identity morphism** $1_m \in \mathbf{Obs}(m,m)$ if for any set of real numbers $\{x_i\}_i$, indexed by $\{1,2,\ldots,m\}$, we have

$$\{\{x_1,\ldots,x_m\},\{x_1,\ldots,x_m\}\} \in 1_m$$

This morphism acts as the **identity** of compositions of morphisms from **Definition 17**.

4.3 Proving that Obs is a Category

Proof of Associativity

Proof. We want to prove that the composition of two morphisms from **Obs**, defined in **Definition 17**, is **associative**:

$$\mathbf{Obs}(m,n) \times (\mathbf{Obs}(l,m) \times \mathbf{Obs}(k,l)) = (\mathbf{Obs}(m,n) \times \mathbf{Obs}(l,m)) \times \mathbf{Obs}(k,l) \quad , \tag{3}$$

for $k,l,m,n \in \mathbf{Obs}$.

Let $f \in \mathbf{Obs}(k,l)$, $g \in \mathbf{Obs}(l,m)$, and $h \in \mathbf{Obs}(m,n)$. We want to prove that

$$h \circ (g \circ f) = (h \circ g) \circ f \quad . \tag{4}$$

Start with the composition from left side, $h \circ (g \circ f)$: by **Definition 17**, the **composition** $h \circ (g \circ f)$ is the set of

$$\{\{x_1 \,\ldots\, x_k\}\ ,\ \{z_1 \,\ldots\, z_n\}\} \in \mathbb{R}^k \times \mathbb{R}^n$$

if there exists a set $\underline{\{y_1 \, \dots \, y_m\} \in \mathbb{R}^m}$ such that

$$\{\{x_1 \dots x_k\}\,,\{y_1 \dots y_m\}\} \in g \circ f \quad \text{and} \quad \{\{y_1 \dots y_m\}\,,\{z_1 \dots z_n\}\} \in h$$

By **Definition 17** for the **composition** $g \circ f$,

$$\{\{x_1 \dots x_k\}\,,\{y_1 \dots y_m\}\} \in g \circ f$$

if there exists a set $\underline{\{w_1 \, \dots \, w_l\} \in \mathbb{R}^l}$ such that

$$\{\{x_1 \dots x_k\}\,,\{w_1 \dots w_l\}\} \in f \quad \text{and} \quad \{\{w_1 \dots w_l\}\,,\{y_1 \dots y_m\}\} \in g$$

We have the following pairs together:

$\{\{x_1 \dots x_k\}\,,\{w_1 \dots w_l\}\} \in f$, $\{\{w_1 \dots w_l\}\,,\{y_1 \dots y_m\}\} \in g$ and $\{\{y_1 \dots y_m\}\,,\{z_1 \dots z_n\}\} \in h$

Summarising, the morphism $h \circ g \circ f$ consist of all pairs

$$\{\{x_1 \dots x_k\}\,,\{z_1 \dots z_n\}\} \in h \circ g \circ f \;\Rightarrow\; h \circ g \circ f \in \mathbf{Obs}(k,n) \tag{5}$$

for which the sets $\{y_1 \dots y_m\} \in \mathbb{R}^m$ and $\{w_1 \dots w_l\} \in \mathbb{R}^l$ do exist.

Continue now with the composition from right side, $(h \circ g) \circ f$: by **Definition 17**, the **composition** $h \circ (g \circ f)$ is the set of all pairs of sets

$$\{\{x_1 \dots x_k\}\,,\{z_1 \dots z_n\}\} \in \mathbb{R}^k \times \mathbb{R}^n$$

if there exists a set $\underline{\{v_1 \, \dots \, v_l\} \in \mathbb{R}^l}$ such that

$$\{\{x_1 \dots x_k\}\,,\{v_1 \dots v_l\}\} \in f \quad \text{and} \quad \{\{v_1 \dots v_l\}\,,\{z_1 \dots z_n\}\} \in h \circ g$$

By **Definition 17** for the **composition** $h \circ g$,

$$\{\{v_1 \dots v_l\}\,,\{z_1 \dots z_n\}\} \in h \circ g$$

if there exists a set $\underline{\{s_1 \, \dots \, s_m\} \in \mathbb{R}^m}$ such that

$$\{\{v_1 \dots v_l\}\,,\{s_1 \dots s_m\}\} \in g \quad \text{and} \quad \{\{s_1 \dots s_m\}\,,\{z_1 \dots z_n\}\} \in h$$

We have the following pairs together:

$\{\{x_1 \dots x_k\}\,,\{w_1 \dots w_l\}\} \in f$, $\{\{v_1 \dots v_l\}\,,\{s_1 \dots s_m\}\} \in g$ and $\{\{s_1 \dots s_m\}\,,\{z_1 \dots z_n\}\} \in h$

Summarising, the morphism $h \circ g \circ f$ consist of all pairs

$$\{\{x_1 \dots x_k\}\,,\{z_1 \dots z_n\}\} \in h \circ g \circ f \;\Rightarrow\; h \circ g \circ f \in \mathbf{Obs}(k,n) \tag{6}$$

for which the sets $\{v_1 \dots v_l\} \in \mathbb{R}^l$ and $\{s_1 \dots s_m\} \in \mathbb{R}^m$ do exist.

The equations (5) and (6) prove (4). □

Proof of the Identity Morphism

Proof. We want to prove that the identity morphism from **Obs**, which was defined in **Definition 18**, is the identity for the composition from **Definition 17**.

For each $m \in ob(\mathbf{Obs})$ we have an **identity morphism** $1_m \in \mathbf{Obs}(m, m)$ if for any set of real numbers $\{x_i\}_i$, indexed by $\{1, 2, \ldots, m\}$, we have

$$\{\{x_1, \ldots, x_m\}, \{x_1, \ldots, x_m\}\} \in 1_m$$

Let $p, q \in ob(\mathbf{Obs})$ and f a morphism from $\mathbf{Obs}(p, q)$. For p, there is an identity morphism 1_p defined as above, i.e., we have a set $\{x_1, \ldots, x_p\}$ that

$$\{\{x_1, \ldots, x_p\}, \{x_1, \ldots, x_p\}\} \in 1_p$$

f a morphism from $\mathbf{Obs}(p, q)$, so we have pairs

$$\{\{x_1, \ldots, x_p\} \ , \ \{y_1, \ldots, y_q\}\} \in f$$

This means that

$$\{\{x_1, \ldots, x_p\}, \{x_1, \ldots, x_p\}\} \in 1_p \text{ and } \{\{x_1, \ldots, x_p\} \ , \ \{y_1, \ldots, y_q\}\} \in f$$

for $\{\{x_1, \ldots, x_p\}, \{x_1, \ldots, x_p\}\} \in 1_p$, so

$$\{\{x_1, \ldots, x_p\} \ , \ \{y_1, \ldots, y_q\}\} \in f \circ 1_p$$

So $f = f \circ 1_p$. Similarly, $1_p \circ f = f$. This concludes the proof that an identity morphism satisfies the conditions of being the identity for the composition of morphisms. □

4.4 The Functor between $\mathcal{W}$ and Obs

We can then define a functor F between the category of flow networks and the category of observations so that for every network in the domain of F we can retrieve the flow information of the network:

$$F : \mathcal{W} \to \mathbf{Obs}$$

Definition 19. This functor consists of:

(a) A function between the objects of the categories $ob(\mathcal{W}) \to ob(\mathbf{Obs})$, where for an item $m \in \mathcal{W}$ we have $m \mapsto \boldsymbol{F}(m) \in ob(\mathbf{Obs})$. This maps a natural number to a natural number.

(b) Taking two objects $m, n \in ob(\mathcal{W})$, then

$$\mathcal{W}(m, n) \mapsto \mathbf{Obs}(\boldsymbol{F}(m), \boldsymbol{F}(n))$$

Observation 20. In the definition of the functor $\boldsymbol{F}$, i.e. **Definition 19**, we need to explain the action of the functor on the set of morphisms $\mathcal{W}(m,n)$ because $\mathcal{W}(m,n)$ is a set of morphisms (not just a single morphism).

The map

$$\mathcal{W}(m,n) \mapsto \mathbf{Obs}(\boldsymbol{F}(m), \boldsymbol{F}(n))$$

means that any morphism from $\mathcal{W}(m,n)$ maps into a morphism of $\mathbf{Obs}(\boldsymbol{F}(m), \boldsymbol{F}(n))$, so we need to check if it is independent of the choice of morphisms.

Observation 21. We then need the axioms of our functor to satisfy:

(c) For any objects $m, n, p \in ob(\mathcal{W})$, the composition of two morphisms $f \in \mathcal{W}(n,p)$, $f' \in \mathcal{W}(m,n)$ is

$$f \circ f' \in \mathcal{W}(m,p)$$

For these objects $m, n, p \in ob(\mathcal{W})$, we have $\boldsymbol{F}(m), \boldsymbol{F}(n), \boldsymbol{F}(p) \in ob(\mathbf{Obs})$. Then the composition $f \circ f' \in \mathcal{W}(m,p)$ gets mapped to the composition in **Obs** via the functor $\boldsymbol{F}$ in the following way:

$$\boldsymbol{F}(f \circ f') = \boldsymbol{F}(f) \circ \boldsymbol{F}(f') \quad , \quad \boldsymbol{F}(f) \in \mathbf{Obs}(\boldsymbol{F}(n), \boldsymbol{F}(p))\, , \; \boldsymbol{F}(f') \in \mathbf{Obs}(\boldsymbol{F}(m), \boldsymbol{F}(n))$$

We explain this further in the proof below.

(d) For any object $m \in ob(\mathcal{W})$ we have an identity $1_m \in \mathcal{W}(m,m)$. Then we have an object $\boldsymbol{F}(m) \in ob(\mathbf{Obs})$ and the corresponding identity $1_{\boldsymbol{F}(m)} \in \mathbf{Obs}(\boldsymbol{F}(m), \boldsymbol{F}(m))$. Then

$$\boldsymbol{F}(1_m) = 1_{\boldsymbol{F}(m)}$$

4.5 Proving the Axioms of the Functor Between $\mathcal{W}$ and Obs

First, we are making the **Definition 19 (b)** more precise: how do we send a morphism from $\mathcal{W}(m,n)$ to a morphism from $\mathbf{Obs}(\boldsymbol{F}(m), \boldsymbol{F}(n))$ (**Observation** 20).

Second, we want to prove the statements in **Observation 21**, that complements the properties of our functor.

4.5.1 Explaining the Second Axiom of the Functor Between $\mathcal{W}$ and Obs

Explaining **Observation** 20.

We stated in **Definition 19 (b)** that any for two objects $m, n \in ob(\mathcal{W})$, then

$$\mathcal{W}(m,n) \mapsto \mathbf{Obs}(\boldsymbol{F}(m), \boldsymbol{F}(n))$$

More rigorously, pick a morphism from $\mathcal{W}(m,n)$: this is a weighted network $(V, E, s, t, S, T, \mathbf{c})$ with m "entries" and n "exits" such that:

- V is the set of vertices of the network.
- E is the set of edges of the network.

- A function s that maps an edge to the vertex which is at the head of the edge.
- A function t that maps an edge to the vertex which is at the tail of the edge.
- S is a subset of V, called **entry vertices**, endowed with a specified total order such that $|S| = m$.
- T is a subset of V, called **exit vertices**, endowed with a specified total order such that $|T| = n$.
- S and T are disjoint non-empty subsets of V.
- $\mathbf{c} : E \to \mathbb{R}$ is a function that maps an edge to its corresponding weight.

Let $S = \{s_1, s_2, \ldots, s_m\}$ and $T = \{t_1, t_2, \ldots, t_n\}$ the sets of entry vertices and exit vertices, respectively.

The functor $\boldsymbol{F}$ send objects from S to the following set, by **Definition 19 (a)**:

$$\{s_1, s_2, \ldots, s_m\} \mapsto \{\boldsymbol{F}(s_1), \boldsymbol{F}(s_2), \ldots, \boldsymbol{F}(s_m)\}$$

and functor $\boldsymbol{F}$ send objects from T to the following set, by **Definition 19 (a)**:

$$\{t_1, t_2, \ldots, t_n\} \mapsto \{\boldsymbol{F}(t_1), \boldsymbol{F}(t_2), \ldots, \boldsymbol{F}(t_n)\}$$

Then

$$\{\{\boldsymbol{F}(s_1), \boldsymbol{F}(s_2), \ldots, \boldsymbol{F}(s_m)\}, \{\boldsymbol{F}(t_1), \boldsymbol{F}(t_2), \ldots, \boldsymbol{F}(t_n)\}\}$$

is a subset of $\mathbb{R}^m \times \mathbb{R}^n$, hence a morphism in $\mathbf{Obs}(\boldsymbol{F}(m), \boldsymbol{F}(n))$.

4.5.2 Proving the Properties Following the Axiom of the Functor Between $\mathcal{W}$ and Obs

Here we want to prove **Observation 21**.

Proving Observation 21 (c): for any objects $m, n, p \in ob(\mathcal{W})$, the composition of two morphisms $f \in \mathcal{W}(n, p)$, $f' \in \mathcal{W}(m, n)$ is

$$f \circ f' \in \mathcal{W}(m, p)$$

For these objects $m, n, p \in ob(\mathcal{W})$, we have $\boldsymbol{F}(m), \boldsymbol{F}(n), \boldsymbol{F}(p) \in ob(\mathbf{Obs})$. Then the composition $f \circ f' \in \mathcal{W}(m, p)$ gets mapped to the composition in **Obs** via the functor $\boldsymbol{F}$ in the following way:

$$\boldsymbol{F}(f \circ f') = \boldsymbol{F}(f) \circ \boldsymbol{F}(f') \quad , \quad \boldsymbol{F}(f) \in \mathbf{Obs}(\boldsymbol{F}(n), \boldsymbol{F}(p)) \, , \, \boldsymbol{F}(f') \in \mathbf{Obs}(\boldsymbol{F}(m), \boldsymbol{F}(n))$$

Proof. Let $m, n, p \in ob(\mathcal{W})$. Take two morphisms from $\mathcal{W}$:

$$(V_1, E_1, s_1, t_1, S_1, T_1, c_1) \in \mathcal{W}(m, n) \quad \text{and} \quad (V_2, E_2, s_2, t_2, S_2, T_2, c_2) \in \mathcal{W}(n, p)$$

The composition of these two morphisms is a morphism in $\mathcal{W}(m, p)$:

$$(V_3, E_3, s_3, t_3, S_3, T_3, c_3) \in \mathcal{W}(m, p) \tag{7}$$

where

$$V_3 = (V_1 \sqcup V_2) \setminus \{(x,1) \mid x \in T_1\} \quad , \quad E_3 = E_1 \cup E_2 \quad , \quad S_3 = S_1 \quad , \quad T_3 = T_2 \tag{8}$$

First, for $(V_1, E_1, s_1, t_1, S_1, T_1, c_1) \in \mathcal{W}(m,n)$,

$$\mid S_1 \mid = m \quad , \quad \mid T_1 \mid = n \quad , \quad S_1 \cap T_1 = \emptyset$$

Let $S_1 = \{s_{11}, s_{12}, \ldots, s_{1m}\}$ and $T_1 = \{t_{11}, t_{12}, \ldots, t_{1n}\}$ the sets of entry vertices and exit vertices, respectively.

The functor $\boldsymbol{F}$ send objects from S_1 to the following set, by **Definition 19 (a)**:

$$\{s_{11}, s_{12}, \ldots, s_{1m}\} \mapsto \{\boldsymbol{F}(s_{11}), \boldsymbol{F}(s_{12}), \ldots, \boldsymbol{F}(s_{1m})\}$$

and functor $\boldsymbol{F}$ send objects from T_1 to the following set, by **Definition 19 (a)**:

$$\{t_{11}, t_{12}, \ldots, t_{1n}\} \mapsto \{\boldsymbol{F}(t_{11}), \boldsymbol{F}(t_{12}), \ldots, \boldsymbol{F}(t_{1n})\}$$

Then

$$\{\{\boldsymbol{F}(s_{11}), \boldsymbol{F}(s_{12}), \ldots, \boldsymbol{F}(s_{1m})\}, \{\boldsymbol{F}(t_{11}), \boldsymbol{F}(t_{12}), \ldots, \boldsymbol{F}(t_{1n})\}\} \tag{9}$$

is a subset of $\mathbb{R}^m \times \mathbb{R}^n$, hence a morphism in $\mathbf{Obs}(\boldsymbol{F}(m), \boldsymbol{F}(n))$.

Second, for $(V_2, E_2, s_2, t_2, S_2, T_2, c_2) \in \mathcal{W}(n,p)$,

$$\mid S_2 \mid = n \quad , \quad \mid T_2 \mid = p \quad , \quad S_2 \cap T_2 = \emptyset$$

Let $S_2 = \{s_{21}, s_{22}, \ldots, s_{2n}\}$ and $T_2 = \{t_{21}, t_{22}, \ldots, t_{2p}\}$ the sets of entry vertices and exit vertices, respectively.

The functor $\boldsymbol{F}$ send objects from S_2 to the following set, by **Definition 19 (a)**:

$$\{s_{21}, s_{22}, \ldots, s_{2n}\} \mapsto \{\boldsymbol{F}(s_{21}), \boldsymbol{F}(s_{22}), \ldots, \boldsymbol{F}(s_{2n})\}$$

and functor $\boldsymbol{F}$ send objects from T_2 to the following set, by **Definition 19 (a)**:

$$\{t_{21}, t_{22}, \ldots, t_{2p}\} \mapsto \{\boldsymbol{F}(t_{21}), \boldsymbol{F}(t_{22}), \ldots, \boldsymbol{F}(t_{2p})\}$$

Then

$$\{\{\boldsymbol{F}(s_{21}), \boldsymbol{F}(s_{22}), \ldots, \boldsymbol{F}(s_{2n})\}, \{\boldsymbol{F}(t_{21}), \boldsymbol{F}(t_{22}), \ldots, \boldsymbol{F}(t_{2p})\}\} \tag{10}$$

is a subset of $\mathbb{R}^n \times \mathbb{R}^p$, hence a morphism in $\mathbf{Obs}(\boldsymbol{F}(n), \boldsymbol{F}(p))$.

For the morphism (7) in $\mathcal{W}(m,p)$:

$$(V_3, E_3, s_3, t_3, S_3, T_3, c_3) = (V_3, E_3, s_3, t_3, S_1, T_2, c_3) \in \mathcal{W}(m,p)$$

we have, according to (8):

$$| S_1 |= m \quad , \quad | T_2 |= p \quad , \quad S_3 \cap T_3 = S_1 \cap T_2 = \emptyset$$

We had $S_1 = \{s_{11}, s_{12}, \ldots, s_{1m}\}$ and $T_2 = \{t_{21}, t_{22}, \ldots, t_{2p}\}$ the sets of entry vertices and exit vertices, respectively.

The functor $\boldsymbol{F}$ send objects from S_1 to the following set, by **Definition 19 (a)**:

$$\{s_{11}, s_{12}, \ldots, s_{1m}\} \mapsto \{\boldsymbol{F}(s_{11}), \boldsymbol{F}(s_{12}), \ldots, \boldsymbol{F}(s_{1m})\}$$

and functor $\boldsymbol{F}$ send objects from T_2 to the following set, by **Definition 19 (a)**:

$$\{t_{21}, t_{22}, \ldots, t_{2p}\} \mapsto \{\boldsymbol{F}(t_{21}), \boldsymbol{F}(t_{22}), \ldots, \boldsymbol{F}(t_{2p})\}$$

Then

$$\{\{\boldsymbol{F}(s_{11}), \boldsymbol{F}(s_{12}), \ldots, \boldsymbol{F}(s_{1m})\}\, , \, \{\boldsymbol{F}(t_{21}), \boldsymbol{F}(t_{22}), \ldots, \boldsymbol{F}(t_{2p})\}\}$$

is a subset of $\mathbb{R}^m \times \mathbb{R}^p$, hence a morphism in $\mathbf{Obs}(\boldsymbol{F}(m), \boldsymbol{F}(p))$.

Therefore, we have

$$\{\{\boldsymbol{F}(s_{11}), \boldsymbol{F}(s_{12}), \ldots, \boldsymbol{F}(s_{1m})\}\, , \, \{\boldsymbol{F}(t_{21}), \boldsymbol{F}(t_{22}), \ldots, \boldsymbol{F}(t_{2p})\}\}$$

is a morphism in $\mathbf{Obs}(\boldsymbol{F}(m), \boldsymbol{F}(p))$ if there exists the set of flow values

$$\{\boldsymbol{F}(s_{21}), \boldsymbol{F}(s_{22}), \ldots, \boldsymbol{F}(s_{2n})\} \in \mathbb{R}^n$$

such that

$$\{\{\boldsymbol{F}(s_{11}), \boldsymbol{F}(s_{12}), \ldots, \boldsymbol{F}(s_{1m})\}\, , \, \{\boldsymbol{F}(s_{21}), \boldsymbol{F}(s_{22}), \ldots, \boldsymbol{F}(s_{2n})\}\} \in \mathbf{Obs}(\boldsymbol{F}(m), \boldsymbol{F}(n)) \quad (11)$$

and

$$\{\{\boldsymbol{F}(s_{21}), \boldsymbol{F}(s_{22}), \ldots, \boldsymbol{F}(s_{2n})\}\, , \, \{\boldsymbol{F}(t_{21}), \boldsymbol{F}(t_{22}), \ldots, \boldsymbol{F}(t_{2p})\}\} \in \mathbf{Obs}(\boldsymbol{F}(n), \boldsymbol{F}(p)) \quad (12)$$

We have that **equation (11)** is shown by **equation (9)** and **equation (12)** is shown by **equation (10)**.

This proves

$$\mathbf{Obs}(\boldsymbol{F}(m), \boldsymbol{F}(p)) = \mathbf{Obs}(\boldsymbol{F}(n), \boldsymbol{F}(p)) \times \mathbf{Obs}(\boldsymbol{F}(m), \boldsymbol{F}(n))$$

So composition of morphisms in $\mathcal{W}$ is mapped by the functor $\boldsymbol{F}$ to the composition of observations in **Obs**. □

Proving Observation 21 (d): For any object $m \in ob(\mathcal{W})$ we have an identity $1_m \in \mathcal{W}(m, m)$. Then we have an object $\boldsymbol{F}(m) \in ob(\mathbf{Obs})$ and the corresponding identity $1_{\boldsymbol{F}(m)} \in \mathbf{Obs}(\boldsymbol{F}(m), \boldsymbol{F}(m))$. Then

$$\boldsymbol{F}(1_m) = 1_{\boldsymbol{F}(m)}$$

Proof. Let $m \in ob(\mathcal{W})$. The identity is then a weighted network

$$1_m = (V, \emptyset, s, t, S, S) \in \mathcal{W}(m, m) \quad ,$$

where $S = T = V$ and $E = \emptyset$ (as shown in **Figure 10** and defined in **Definition 13**). Recall that we identify the entry vertices where the flows come in, S, with the exit vertices where the flow goes out, T, because there are no edges between S and T. Both S and T are the size of V, because the identity has no other vertices (free of flow-in or flow-out).

Then

$$\mid S \mid = m \quad \text{with} \quad S = \{s_1, s_2, \dots s_m\} \quad (= T)$$

The functor $\boldsymbol{F}$ sends s_i to $\boldsymbol{F}(s_i) \in ob(\mathbf{Obs})$:

$$S = \{s_1, s_2, \dots s_m\} \mapsto \{\boldsymbol{F}(s_1), \boldsymbol{F}(s_2), \dots \boldsymbol{F}(s_m)\}$$

and, so does with T, since $T = S$. Then we have the observation

$$\{\{\boldsymbol{F}(s_1), \boldsymbol{F}(s_2), \dots \boldsymbol{F}(s_m)\} \, , \{\boldsymbol{F}(s_1), \boldsymbol{F}(s_2), \dots \boldsymbol{F}(s_m)\}\}$$

in $\mathbf{Obs}(m, m)$, which is our identity morphism in category $\mathbf{Obs}$:

$$1_{\boldsymbol{F}(m)} = \{\{\boldsymbol{F}(s_1), \boldsymbol{F}(s_2), \dots \boldsymbol{F}(s_m)\} \, , \{\boldsymbol{F}(s_1), \boldsymbol{F}(s_2), \dots \boldsymbol{F}(s_m)\}\}$$

So, identity morphism in $\mathcal{W}$ is mapped by $\boldsymbol{F}$ into the identity morphism in $\mathbf{Obs}$. □

4.6 Where is the Equivalence?

The categorisation of weighted networks $\mathcal{W}$ and of observation **Obs** has been proven previously and it is because of the functor defined in **definition 19** that we can give the concept of flow to its respective specific weighted capacitance network. With the functor, it is now possible to extract observations from a weighted network. Then we can define two flow networks to be equivalent if we have the same observation of flow:

Let $m, n \in ob(\mathcal{W})$ and $f, f' \in \mathcal{W}(m, n)$ two weighted networks with m vertices of entries and n vertices of exists, then we say that the networks f and f' are equivalent if $\mathbf{F}(f) = \mathbf{F}(f')$, i.e their observations are equal.

5 Conclusion

5.1 Evaluation

In this thesis, we focused on flow networks, a class of directed graphs from a source to a sink.

Previously the challenge of maximum flow within a network was dealt with graph theory and resolved using linear programs. As of today, the concepts and techniques of discrete mathematics are enriched with the novel frameworks of category theory; we applied this framework to flow networks. Consequently, we analyzed and decomposed these mathematical structures using directed graphs, where nodes represented objects (or sets), and the vertices represented the functions from an object to another. This thesis was organized to go from a small mathematical structure, directed graphs, to a bigger one, a weighted network, through the notion of simple networks to make sure that the different categories along the way are well defined and also to simplify proofs when applicable. These categories rigorously constructed, we turned our attention to categorize the meaning of flows using a category theoretical approach, doing so through observations. Finally, attaching the notion of flow to the category of weighted networks using functors, we could establish specifically an equality between categories of flow networks. The main ideas that supported the mathematical research process were those given by work on networks, and we took great care to set out properly and correctly the properties of the network. Finally, we were able to perform a good categorization of networks built upon its properties, something that facilitated the work needed to determine a flow network.

The main ideas that supported the mathematical research process were those given by work on networks, such as John Baez [Bae+20] as the thesis would have been invalid if the properties of the network were not set out properly and correctly. A good categorization of networks built upon its properties then facilitates the work needed to determine a flow network.

In our thesis, the most important demonstrations were the ones required for the category of observations. In effect, without a properly defined category that satisfied the axioms needed, no functor would have been able to be created to relate a flow to a network, thus rendering a weighted network useless.

In the end, despite the main problems raised during the research process (having to deal with an infinity of maximum flows), the decisions taken (having to create the category of observations necessary to introduce the notion of flow), we hope that other researchers will find rigorous enough materials in this thesis to establish the presentation of the category of flow networks. Equipped with it, hopefully they will ultimately be able to construct a generator of optimized flow networks which, according to the lifelong dedication from S. MAC LANE and S. EILENBERG, would significantly unify, clarify and render efficient the complexity of the systems in question.

5.2 Further Directions

The next step in this thesis would be to prove the monoidality of the flow network that we have defined. This would show that we can compose the networks not only horizontally but also vertically, by applying a tensor on two morphisms f and g to form $f \otimes g$ as described by John Baez [bae16]. This would mean being able to construct more intricate networks than just a line of networks.

After proving the monoidalitic properties of the flow network, similarly to Yves Lafont [Laf03] with boolean circuits, we would transition to the presentation of the category of flow network. Building a presentation means finding a set of generators and a set of relations between the generators. This means that with a presentation, we would be able to generate any flow networks, a step that is much needed if we were working with optimization problem such as finding the maximum flow.

After having found the presentation, a possible area to generalize further would be to construct a subcategory of optimal flow networks with constraints. This would extend the category of flow network to then categorize what an optimal flow in a network represents, a result which could then be coupled with rewriting theory and applied on notions of complexity theory in relations to maximum flow.

Finally, to complete the main proof of this thesis, we had to restrict it to the networks that are observable to be able to use a category of observation. Removing this limitation would generalize the results to the networks that are not observable. It doesn't change the use of the result in practice but makes this proof a more robust stepping stone for future work in the field.

Bibliography

[For56] L.R Ford. *Network Flow Theory.* The RAND Corporation, 1956.

[FF56] L.R Ford and D.R. Fulkerson. *Notes on Linear Programming - part 32: Solving the Transportation Problem.* Flow in Networks. The RAND Corporation, 1956.

[BG61] Robert Busacker and Paul Gowen. "A procedure for determining a family of minimum cost network flow patterns". In: *Technical Report 15, Operation Research, Johns Hopkins University* (1961).

[ML71] Saunders Mac Lane. *Categories for the Working Mathematician.* Springer Science + Business Media, 1971.

[Har80a] L Harper. "The global theory of flows in networks". In: *Journal of Advances in Applied Mathematics* 1.2 (1980), pp. 158–181.

[Har80b] L. H. Harper. *The Global Theory of Flows in Networks.* Academic Press. Inc., 1980.

[GKP94] Ronald Graham, Donald Knuth, and Oren Patashnik. *Concrete Mathematics: A Foundation for Computer Science.* 2nd ed. Addison-Wesley Publishing, 1994.

[Bol98] Bela Bollobas. *Modern Graph Theory.* 1st ed. Springer, 1998.

[Laf03] Yves Lafont. "Towards an algebraic theory of Boolean circuits". In: *Journal of Pure and Applied Algebra* 184.2-3 (2003), pp. 257–310. DOI: 10.1016/s0022-4049(03)00069-0.

[Bla04] Paul Black. *flow network.* 2004. URL: https://xlinux.nist.gov/dads/HTML/flownetwork.html.

[KZ04] V. Kolmogorov and R. Zabih. *What Energy Functions Can Be Minimized via Graph Cuts?* 2004.

[Hei] "Chapter 8: Network Flow Algorithms". In: *Algorithms in a Nutshell.* O'Reilly Media, 2008, pp. 226–250.

[Cor+09a] Thomas Cormen et al. *Introduction to Algorithms.* 3rd ed. MIT Press, 2009.

[Cor+09b] Thomas H Cormen et al. *Introduction to algorithms.* 3rd ed. Mit Press, 2009.

[Gal10] Jean Gallier. *Discrete Mathematics.* Springer, 2010.

[SW11] Robert Sedgewick and Kevin Wayne. *Algorithms.* 4th ed. Addison-Wesley Professional, 2011, p. 886.

[LLM12] Eric Lehman, Frank Leighton, and Albert Meyer. *Mathematics for Computer Science.* 2012, pp. 297–300.

[SF13] Robert Sedgewick and Philippe Flajolet. *An introduction to the analysis of algorithms.* 2nd ed. Addison-Wesley, 2013.

[Lei14] Tom Leinster. *Basic category theory.* Cambridge University Press, 2014.

[DH15] Timothy Davis and Yifan Hu. "II.16 Graph Theory". In: *The Princeton Companion to Applied Mathematics.* Ed. by Nicholas Higham. Princeton University Press, 2015, pp. 101–103.

[Hig15a] Nicholas Higham. "I.4 Algorithms". In: *The Princeton Companion to Applied Mathematics.* Ed. by Nicholas Higham. Princeton University Press, 2015, pp. 40–48.

[Hig15b] Nicholas Higham. “I.5 Goals of Applied Mathematical Research”. In: *The Princeton Companion to Applied Mathematics*. Ed. by Nicholas Higham. Princeton University Press, 2015, pp. 48–54.

[Mor15] Esteban Moro. “Section IV.18 Network Analysis”. In: *The Princeton Companion to Applied Mathematics*. Ed. by Nicholas Higham. Princeton University Press, 2015, pp. 360–374.

[Vyg15] Jens Vygen. “IV.38 Combinatorial Optimization”. In: *The Princeton Companion to Applied Mathematics*. Ed. by Nicholas Higham. Princeton University Press, 2015, pp. 564–570.

[Win15] Peter Winkler. “IV.37 Applied Combinatorics and Graph Theory”. In: *The Princeton Companion to Applied Mathematics*. Ed. by Nicholas Higham. Princeton University Press, 2015, pp. 552–564.

[Wri15] Stephen Wright. “IV.11 Continuous Optimization (Nonlinear and Linear Programming)”. In: *The Princeton Companion to Applied Mathematics*. Ed. by Nicholas Higham. Princeton University Press, 2015, pp. 281–293.

[bae16] John baez. 2016. URL: http://math.ucr.edu/home/baez/networks_santa_fe/networks_santa_fe.pdf.

[BI16] Saugata Basu and M. Umut Isik. *Categorical Complexity*. 2016. URL: https://www.math.purdue.edu/~sbasu/categoricalcomplexity-25-oct-2019.pdf.

[Gar+16] Sanjam Garg et al. “Hiding Secrets in Software: A Cryptographic Approach to Program Obfuscation”. In: *Communications of the ACM* 59.5 (2016), p. 116.

[BLP17] Srijan Biswas, Saswata Sundar Laga, and Biswajit Paul. “A Review on Ford Fulkerson Graph Algorithm for Maximum Flow”. In: *International Journal of Scientific Engineering Research* 8.3 (2017), pp. 109–112.

[Rie17] Emily Riehl. *Category theory in context*. Dover Publications, 2017.

[Mil18] Bartosz Milewski. *Category theory for programmers*. 2018.

[ES19] Nasser El-Sherbeny. “The Fuzzy Minimum Cost Flow Problem with the Fuzzy Time-Windows”. In: *American Journal of Applied Mathematics and Statistics* 7.6 (2019), pp. 191–195.

[Mar19] Jean-Pierre Marquis. *Category Theory (Stanford Encyclopedia of Philosophy/Fall 2019 Edition)*. 2019. URL: https://plato.stanford.edu/archives/fall2019/entries/category-theory/.

[Ros19] Kenneth Rosen. *Discrete mathematics and its applications*. 8th ed. McGraw-Hill Education, 2019.

[Wil19] David Williamson. *Network Flow Algorithms*. Cambridge University Press, 2019.

[Bae+20] John Baez et al. *Network Models*. 2020.

[God] Laurent Godefroy. *Chapitre 06 - Recherche d’un flot maximum dans un réseau de transport | SUPINFO, école Supérieure d’Informatique*. URL: https://www.supinfo.com/cours/2GRA/chapitres/06-recherche-flot-maximum-reseau-transport.