FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Weighted coupled cell networks and invariant synchrony patterns

Pedro Manuel Nunes Sequeira

Doctoral Program in Electrical and Computer Engineering (PDEEC)

Supervisor: António Pedro Rodrigues Aguiar

Co-Supervisor: João Pedro Hespanha

20 June 2023

Weighted coupled cell networks and invariant synchrony patterns

Pedro Manuel Nunes Sequeira

Doctoral Program in Electrical and Computer Engineering (PDEEC)

20 June 2023

Resumo

A primeira grande contribuição desta tese é o desenvolvimento de um formalismo baseado em monóides comutativos para o estudo de redes de células acopladas ponderadas. Este formalismo generaliza o anterior de forma a lidar com arestas ponderadas arbitrárias, e desenvolve o conceito de componente oráculo, que é um objecto matemático que descreve como as células de um determinado tipo respondem a vizinhanças finitas arbitrárias. Isto separa completamente a modelação do comportamento das células, da rede particular onde as células de interesse se inserem. Para além disso, permite construir funções admissíveis, que podem ser usadas para modelar a dinâmica de uma rede, ou alguma função de medição. Mostramos também que este formalismo pode ser trivialmente estendido para redes com inputs exógenos e células com parâmetros internos.

A segunda grande contribuição é a prova matemática que os resultados conhecidos sobre partições balanceadas e as suas respectivas latices também se aplicam nesta situação mais geral. Para além disso, muitos destes resultados podem ser estendidos para padrões de sincronismo gerais (baseados em igualdades).

Além disso, o algoritmo *coarsest invariant refinement* (CIR) para encontrar partições balanceadas é generalizado para redes ponderadas e a sua performance é melhorada.

A terceira grande contribuição é o estudo da influência da estrutura da rede no comportamento qualitativo de conjuntos de sincronismo invariante, em particular, em relação aos diferentes tipos de in-vizinhanças (cumulativas) e conjuntos in-alcançáveis. Isto motiva a classificação das partições nas categorias de *strong*, *rooted* e *weak*, de acordo com sua relação com a estrutura de conectividade da rede.

Por fim, a quarta principal contribuição foi tornar explícitos os graus de liberdade envolvidos no projeto de uma componente oráculo.

Abstract

The first major contribution of this thesis is the development of a framework based on commutative monoids for analyzing weighted coupled cell networks. This framework generalizes a previous formalism in order to deal with arbitrary weighted edges. Furthermore, it develops the concept of oracle component, which is a mathematical object that describes how cells of a given type respond to arbitrary finite in-neighborhoods, and it completely separates the modeling of the behavior of cells from the particular network on which the cells of interest are inserted. This allows us to construct admissible functions, which can be used to model the dynamics of a network, or some measurement function. We also show how this formalism can be trivially extended to networks with exogenous inputs and cells with internal parameters.

The second major contribution is to show that the known results about balanced partitions and their respective lattices also apply to this more general setup. Furthermore, many of these results can be extended to general (equality-based) invariant synchrony patterns.

Additionally, the coarsest invariant refinement (CIR) algorithm to find balanced partitions is generalized for weighted networks and its performance is improved.

The third major contribution is the analysis of the influence of the structure of the network in the qualitative behavior of invariant synchrony sets, in particular, with respect to the different types of (cumulative) in-neighborhoods and the in-reachability sets. This motivates the classification of the partitions into the categories of strong, rooted and weak, according to their relation to the connectivity structure of the network.

Finally, the forth main contribution is to make explicit the degrees of freedom involved in the design of an oracle component.

Acknowledgments

I would firstly like to thank my supervisors António Pedro Aguiar and João Pedro Hespanha for their guidance.
I am also thankful to the University of California, Santa Barbara for allowing me to be a visitor during part of my program.
Lastly, I am grateful to the many amazing tools that available in my age. In particular, to Wikipedia and to "The On-Line Encyclopedia of Integer Sequences" (OEIS). Also, the Youtube channels "Numberphile", "3Blue1Brown" and "The Bright Side of Mathematics" are partly responsible for my mathematical growth and keeping my brain sharp.

This work was supported by a Ph.D. Scholarship, grant SFRH/BD/119835/2016 from Fundação para a Ciência e a Tecnologia (FCT), Portugal (POCH program).

Pedro Sequeira

"There is nothing more practical than a good theory."

Kurt Lewin

Contents

1 Introduction **1**
1.1 Outline and main contributions 4

2 Weighted CCN formalism **5**
2.1 Commutative monoids 7
2.2 Multi-indexes 10
2.3 Weighted coupled cell networks 10
2.4 Admissibility 11
2.5 Extension for exogenous inputs and inner cell parameters 18

3 Equality-based synchronism **21**
3.1 Partitions and their representations 21
3.2 Lattices of partitions 25
3.3 Lattice quotients 29
3.4 Polydiagonals 31
3.5 Invariance of polydiagonals 33
3.6 Balanced partitions 35
3.7 Quotient networks 40
3.8 CIR algorithm for balanced partitions 45
3.8.1 Method 45
3.8.2 Efficient implementation and cost analysis 46

4 In-reachability based classification of synchrony partitions **51**
4.1 Network connectivity 51
4.1.1 Neighborhoods and reachability 51
4.1.2 Dynamics from in-neighborhoods 53
4.1.3 Strongly connected components and root dependency 55
4.2 Strong, rooted and weak partitions 56
4.3 Neighborhood color matching 60
4.4 Neighborhood color invariance 64

5 Output vector spaces **73**
5.1 Admissible vector spaces and related results 73
5.2 Decomposition into coupling components 77
5.3 Decomposition into basis components 93
5.3.1 Multiplicity notation 93
5.3.2 Stirling numbers 94
5.3.3 Finite coupling order 95

5.3.4 Infinite coupling order 109
5.4 Extension for exogenous inputs and inner cell parameters 111

6 Conclusion **113**

References **115**

A Intermediate results **117**
A.1 Intermediate results used in Theorem 5.3.5 117
A.2 Intermediate results used in Theorem 5.3.24 127

List of Figures

2.1 A weighted coupled cell network. 5
2.2 Logical gates. 6
2.3 Example of a cell in a discrete-time system. 6
2.4 Example of a continuous-time network. 7
2.5 Edge merging. 11
2.6 Simple network with admissible functions that have the structure given by Equations (2.7) to (2.9). 15
2.7 Network with inner cell parameters and exogenous inputs. 18

3.1 Partially ordered sets L, S such that L is a lattice and S is not a lattice. 26
3.2 Illustration of a cir_L function over a suitable lattice of partitions L. 28
3.3 Illustration of a lattice L and its quotient lattice $L/\mathcal{A}$. 31
3.4 Networks with equivalent input sets. 38
3.5 Color-coded network of Figure 2.6 and its quotient over the balanced partition $\{\{1,2\},\{3\}\}$. 42
3.6 Chain **CCN**. 43
3.7 Unbalanced coloring. 43
3.8 Network of Example 3.8.6 illustrating the CIR algorithm. 47

4.1 Simple chain of 4 cells. 52
4.2 Decomposition of a network into its strongly connected components. 56
4.3 A network and its lattice of balanced partitions. 60
4.4 Partition that is not $\mathcal{N}^-$-matched but is $\mathcal{V}^-$-matched. 62
4.5 Partition that is not $\mathcal{V}^-$-matched but is $\mathcal{R}^-$-matched. 62
4.6 A network and its lattice of balanced partitions. 64
4.7 Example of a spurious (not $\mathcal{N}^-$-invariant) partition. 67
4.8 Example of a partition that is neither $\mathcal{N}^-$-invariant nor $\mathcal{N}^-$-matched. 68
4.9 Example of a partition that is not $\mathcal{N}^-$-invariant but is $\mathcal{V}^-$-invariant. 68
4.10 Example of a partition that is not $\mathcal{V}^-$-invariant but is $\mathcal{R}^-$-invariant. 69
4.11 Example of a quotient over a partition that is not $\mathcal{R}^-$-invariant. 70
4.12 Example of a quotient over a partition that is not $\mathcal{R}^-$-invariant. 70
4.13 Lattices of balanced partitions of a network and its quotients. 71

5.1 Simple input sets. 77

List of Tables

4.1 Join table for general partitions. 59
4.2 Join table when rooted partitions are $\mathcal{R}^-$-matched. 64
4.3 Relation between partitions and their quotients over a $\mathcal{R}^-$-invariant partition. . . 69

Abbreviations and Symbols

CCN	Coupled Cell Network
CIR	Coarsest Invariant Refinement
SCC	Strongly Connected Component
RDC	Root Dependency Component

Chapter 1

Introduction

Networks are structures that describe systems with multiple components, called **cells**. These cells can be pairwise connected through **edges**, which encode how one cell affects another. In general, these edges can be directed or undirected and they can have weights in order to parameterize their interaction.

These are ubiquitous structures, both in the natural world and in engineering applications. Some examples are for instance the brain, the internet, the electric grid and electronic circuits in general, food webs and the spread of a virus in a pandemic.

In order to study these types of systems, the theory of **coupled cell networks** (CCN) was first developed in Stewart et al. (2003); Golubitsky et al. (2005); Golubitsky and Stewart (2006). The formalism used in their work is based on groupoids of bijections between in-neighborhoods of cells.

In the theory of CCNs, the concept of **admissibility** is defined such that a function f is admissible in some network, if it satisfies certain minimal properties that allow it to be a plausible modeling of some dynamical system $\mathbf{x}^+/\dot{\mathbf{x}} = f(\mathbf{x})$ or measurement function $\mathbf{y} = f(\mathbf{x})$ on that network.

These functions have to be "first-order", in the sense that we are modeling something that, when evaluated at a cell, depends on the state of that cell and its in-neighbors. This does not mean that everything on a network has to (or can) be defined by such a function. For instance, the second derivative or the two-step evolution of the aforementioned dynamical systems will not be of this form. Those functions will be "second-order" in the sense that they are dependent on their first and second in-neighborhoods (neighbor of neighbor). They are, however, fully defined from the original first-order functions.

This line of work also introduced the notion of **quotient network**, which is a smaller network that describes the behavior of the original network when the state of a system is in an (equality-based) **invariant synchrony pattern**. This means that some cells are sharing the same state and will continue doing so. Important limitations arose from the fact that this formalism assumed only single edges between each ordered pair of cells. For instance, a quotient network might not satisfy this assumption even if the original network of interest does. This issue was solved by the "multiarrow formalism" developed in Golubitsky et al. (2005), which allows the existence of multiple edges between the same pair of cells and self-loops. This extended the formalism to the simplest weighted case, which consists of integer weights that are used to represent a number of identical "unitary" edges in parallel. This was then further extended in Aguiar et al. (2017), in order to allow for edges that are parameterized by real values, however, only admissible functions with an additive in the weights structure were considered.

In Sequeira et al. (2021), we introduced a formalism for general weighted CCNs, which is a proper generalization of the groupoid formalism. Our formalism uses the algebraic structure of the commutative monoid to deal with arbitrary edge sets. This is the minimal structure, with the necessary symmetry properties, that is able to encode finite edges in parallel.

Much more important than the extension to general weights, is the development of the concept of **oracle components**. An oracle component is a mathematical object that describes how cells of a given type respond to arbitrary finite in-neighborhoods. It completely separates the modeling of cell behavior from the particular network on which the cells are inserted.

This approach is very general in the sense that we only impose the same type of equality constraints used in the original CCN formalism. That is, if two cells of the same type are in the same state and have equivalent in-neighborhoods, then at that instant, they should behave in the same manner. We found it more convenient to specify the notion of in-neighborhood equivalence through items 1 to 3 of Definition 2.4.1, which act as generators of this set of equalities instead of working with a pullback map on a groupoid of bijections, which would be the generated object.

Then, to specify an admissible function on a CCN, which models its dynamics (or an output function in general), we just need to choose a tuple of oracle components (one for each cell type), which is called an **oracle function**. The admissible function is then obtained by evaluating on each cell, together with its corresponding in-neighborhood, the appropriate oracle component.

Note that the oracle component is a much preferable mathematical object to work with than the admissible function. That is, in order to study a function, we would rather know it completely than just knowing its value when evaluated at some points. In particular, despite the fact that in most applications we might not have to deal with cells that have arbitrarily large in-neighborhoods, it proves essential for the oracle components to be properly defined in such cases.

In dynamical systems on networks, the study of synchrony between the different cells is often of the utmost importance Strogatz and Stewart (1993). Some examples are the cardiac pacemaker cells responsible for our heartbeat, the flashing of a swarm of fireflies, the consensus problem in control theory and the different gaits in animal locomotion generated by "central pattern generators" (CPG). There are, however, situations in which too much synchronism is actually undesirable, such as in epileptic seizures in the brain.

One of the most predominant models in the study of synchrony of oscillators is the Kuramoto model, which consists on a large set ($N \to \infty$) of simple oscillators that are weakly coupled in an all-to-all fashion. Some reviews on the Kuramoto and its variants can be found in Arenas et al. (2008); Dörfler and Bullo (2014); Rodrigues et al. (2016). Although there are many variants, many important models are usually given by dynamical functions with very simple structure, such as being "additive in the edges/weights" or being "weakly coupled". Nevertheless, the importance of studying systems with higher order couplings has been recognized Battiston et al. (2021), as reviewed in Bick et al. (2021); Battiston et al. (2020).

We refer the reader to Memmesheimer and Timme (2012), where it is experimentally observed that changing additive coupling dynamics to non-additive can enable persistent synchrony, with this phenomenon appearing even in random networks, with no pre-constructed graph structure that would justify the existence of synchronism. The inability of the additive coupling system to exhibit such a feature might mean that such a system is, in some sense, degenerate.

This has lead to many works that extend the concept of network, such as hypernetworks Aguiar et al. (2022) and simplicial complexes Nijholt and DeVille (2022).

This generalization of networks into more complex, higher dimensional structures is motivated by the objective of constructing admissible functions that have non-pairwise terms. In this work, we show that despite being simpler structures, standard networks are also capable of having higher order, non-pairwise terms, although they are constrained in a very particular way.

There have been some extensions of the Kuramoto model to higher coupling orders Aguiar and Dias (2018); Ashwin and Rodrigues (2016); Bick et al. (2016). While these functions certainty are invariant to permutations (item 1 of Definition 2.4.1) and dependent only on the cell of reference and its in-neighbors (item 3 of Definition 2.4.1), it is not clear whenever they follow the edge-merging principle (item 2 of Definition 2.4.1), which is a very strong constraint. Note that this last condition was already present in the original groupoid formalism and it is essential in order to properly define quotient networks. The weighted formalism used here only makes it more explicit.

In this work, we study general equality-based invariant synchrony patterns, which are represented through partitions on the set of cells of a network. Much work has been done regarding **balanced partitions**, which represent patterns of synchrony that are invariant under any admissible function on the network of interest. Although balanced partitions represent a very important subclass of invariant synchrony patterns with strong properties, such as implying the existence of quotient networks, it is possible for other invariant patterns to be present in a network.

Consider for instance the subset of admissible functions such that a cell becomes insensitive to cells that are on the same state. Note that such a system is, consequently, always insensitive to self-loops. This happens, for instance, in the Kuramoto model. This property leads to the study of exo-balanced partitions Aguiar and Dias (2018); Neuberger et al. (2020); Aguiar and Dias (2021), which is a larger class of partitions than the balanced ones.

For this reason, we consider arbitrary subsets of admissible functions F and show that the set of partitions L_F that describe (equality-based) synchrony patterns that are invariant under F always form lattices. Furthermore, we show that these lattices have similar properties to the lattices of balanced partitions Λ_G Stewart (2007). In particular, these lattices share the same join operation $\vee$ and have a cir_F function associated with them.

The **coarsest invariant refinement** (*cir*), was first developed in Aldis (2008) as a polynomial-time algorithm that finds the maximal element of the lattice of balanced partitions. In Neuberger et al. (2020) it was noted that this algorithm does more that just finding the maximal balanced partition. In fact, given any input partition, it produces the greatest balanced partition that is finer ($\leq$) than

the input one.
We improved and generalized this algorithm for arbitrary weights in Sequeira et al. (2021), which has a worst-case time complexity of $\mathbf{O}(|\mathcal{C}|^3)$ instead of $\mathbf{O}((|\mathcal{E}|+|\mathcal{C}|)^4)$ as in Aldis (2008), where $\mathcal{C}$ and $\mathcal{E}$ denote the sets of cells and edges, respectively.
In Sequeira et al. (2023b), we show that the concept of *cir*, as a function, is not specific to balanced partitions and that every F-invariant lattice L_F has an associated cir_F function. Furthermore, we explore how the connectivity structure of a CCN affects an admissible dynamical system in that network. In particular, we focus on the different types of (cumulative) in-neighborhoods and the in-reachability sets. We show that differences in this structure can lead to qualitatively different behaviors of general equality-based invariant synchrony patterns.
The oracle components are defined through a set of equality constraints to itself. This is a high-level description in the sense that it is not clear how one could construct such an object. In Sequeira et al. (2023a), we dissected these mathematical objects by making explicit their degrees of freedom. For this purpose, we define the concept of **coupling components** and **basis components**.

1.1 Outline and main contributions

In Chapter 2 we introduce the commutative monoid formalism, for general weighted CCNs, which we developed in Sequeira et al. (2021) and then simplified in Sequeira et al. (2023a). This formalism allows us to extend the previous known results about CCNs to networks with weighted connections, with arbitrary amount of edges and edge types. Furthermore, we develop the concept of oracle functions, which allows us to construct the admissible functions associated with any network in a systematic and self-consistent manner. Finally, we show how to easily extend this formalism to networks that allow exogenous inputs and cells with internal parameters.
In Chapter 3 we use the new and more general definitions of admissibility to extend the previous known results about equality-based invariant synchrony patterns. We go beyond the well-known balanced partitions and respective lattice by extending these concepts to the lattices induced by subsets of admissible functions. Furthermore, we show the existence of a *cir* function for these general lattices and we present our improvement of the CIR algorithm for balanced partitions.
In Chapter 4 we clarify how the connectivity structure of a network affects its dynamics. This motivates the study of the network according to its in-reachability sets, which leads to the definition of a classification scheme of partitions of invariant synchrony into strong, rooted and weak types.
In Chapter 5, we focus on the particular case where the output set of the admissible functions is a vector space. Here, we provide results in terms of local robustness that apply for this type of spaces. Furthermore, we introduce two decompositions which make explicit the degrees of freedom involved in the design of oracle components.

Chapter 2

Weighted CCN formalism

In this chapter we describe the formalism for general networks with arbitrary, weighted connections. Furthermore, we show that previous known results about CCNs can be extended to these more general networks. The schematic in Figure 2.1 illustrates the object of study. It represents a

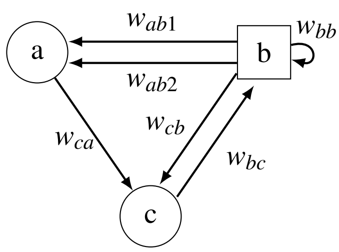
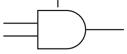


Figure 2.1: A weighted coupled cell network.

network containing three cells $\mathcal{C} = \{a,b,c\}$. Furthermore, cells a and c are both represented with a circle, meaning that they are objects of the same type, while cell b is of a different type. We usually index the set of cell types with integers, such as $T = \{1,2\}$ (e.g., identify "circle" with 1 and "square" with 2). Our goal is to study the global dynamical system associated with a network in which its cells are also dynamical systems in their own right. This means that for each cell type $i \in T$ we have an associated state set $\mathbb{X}_i$ and output set $\mathbb{Y}_i$, which we use to build the domains and co-domains of the associated dynamical systems. A cell can influence the dynamical evolution of other cells in the network, which we represent by a directed edge, where the receiving cell is the affected one. Each interaction can be arbitrarily parameterized by associating a weight/label on the corresponding directed edge. It is possible for a cell to affect another in multiple ways, as seen in Figure 2.1, where cell b affects cell a through an interaction parameterized by w_{ab1} and another one parameterized by w_{ab2}. Using the algebraic structure of the commutative monoid described in the following section, we can represent this by a single interaction parameterized by $w_{ab1} \| w_{ab2}$, which is a common notation used in electrical circuit theory. We now present an example of a possible cell in some digital circuit.

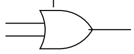
Example 2.0.1. *Consider the system illustrated in Figure 2.3, which is composed by the logical gates described in Figure 2.2. The logical gates have* ***bits*** *as inputs, that is, elements of* $\{0,1\}$ *and*

and	0	1
0	0	0
1	0	1



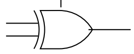
(a) AND logical gate.

or	0	1
0	0	1
1	1	1



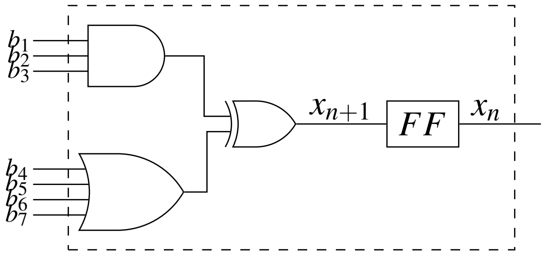
(b) OR logical gate.

xor	0	1
0	0	1
1	1	0

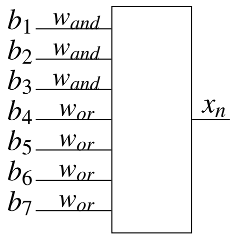


(c) XOR logical gate.

Figure 2.2: Logical gates.



(a) Cell implementation.



(b) Cell seen from the outside.

Figure 2.3: Example of a cell in a discrete-time system.

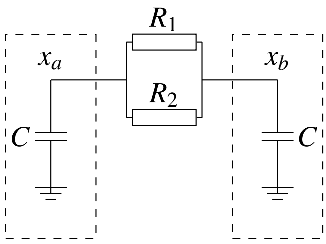
they output another bit. Note that these objects can be extended for arbitrary finite inputs in the natural way. In particular, the AND and OR gates behave as their corresponding logical operators by identifying 0 *and* 1 *with* ***false*** *and* ***true****, respectively. The XOR gate (meaning eXclusive OR) simply acts as a sum modulo* 2*. Finally, the box labeled FF (meaning flip-flop) is a memory unit, which we consider to update its value periodically. Then, the dashed rectangle in Figure 2.3 represents a possible cell in a network with $\mathbb{X}_i = \mathbb{Y}_i = \{0,1\}$, where $i \in T$ represents its cell type. This particular cell has inputs $\{b_1,\dots,b_7\}$, which are bits that can come from other cells in the underlying network. The dynamics of this particular cell is then given by*

$$x_{n+1} = xor(and(b_1,b_2,b_3), or(b_4,b_5,b_6,b_7)).$$

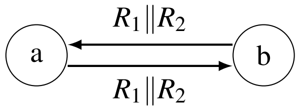
Note that the set of inputs $\{b_1,\dots,b_3\}$ plays a different role than $\{b_4,\dots,b_7\}$ in this cell. Such distinction cannot be seen physically since all inputs affect the cell through similar-looking wires. This is solved by appropriately labeling the wires, in this case with w_{and} and w_{or}. Note that these labels fully describe how a bit through a given wire affects the corresponding cell. This means that as long as the edges (wires) are appropriately labeled and everything is connected accordingly, then, the relative positions of the wires in Figure 2.3b are not important. In summary, if the relative positions were still important that would just mean that the current weights/labels do not provide us with a complete description of how that edge affects the cell. □

We now illustrate how a very simple electric circuit can be represented with this formalism.

Example 2.0.2. *Consider the circuit in Figure 2.4a, with two cells a and b, each consisting of a capacitor C. The state of each cell corresponds to the potential x_a and x_b in the indicated terminals. Here we have $\mathbb{X}_i = \mathbb{Y}_i = \mathbb{R}$, where $i \in T$ is the cell type under consideration. These cells are connected to each other through two resistors R_1 and R_2, or according to the usual*



(a) Electric circuit.



(b) CCN representation.

Figure 2.4: Example of a continuous-time network.

notation, through $R_1\|R_2$, which is read as "R_1 in parallel with R_2". This circuit is represented by the network in Figure 2.4b, where we note that from the fact that resistors are fundamentally bidirectional, each cell has an edge of weight $R_1\|R_2$ coming from the other cell. The dynamics of this particular network are given by

$$\dot{x}_a = \frac{1}{C}\frac{1}{R_1\|R_2}(x_b - x_a),$$
$$\dot{x}_b = \frac{1}{C}\frac{1}{R_1\|R_2}(x_a - x_b),$$

with $R_1\|R_2 = \frac{R_1 R_2}{R_1+R_2}$. □

In the following section we describe the appropriate algebraic structure used in order to encode the connectivity between cells.

2.1 Commutative monoids

The commutative monoid is a set equipped with a binary operation (usually denoted $+$) such that it is commutative and associative. Furthermore, it has one identity element (usually denoted 0). This is the simplest algebraic structure that can be used to describe arbitrary finite parallels of edges. Note that commutativity and associativity, together, are equivalent to the invariance to permutations property. This reflects the fact that, for any given set of edges in parallel, it is irrelevant the order in which we enumerate the individual edges.

In this work, we denote the monoid "sum" operation by $\|$, due to the context in which it is used, with the meaning of "adding in parallel". Nevertheless, it is convenient to think of this as a sum. Likewise, the notation $\sum$ is used to describe parallels of multiple edges. In this context, the zero element of a monoid should be interpreted as "no edge".

Note that we do not require the existence of inverse elements. That is, given an edge, there does not need to exist another one such that the two in parallel act as "no edge". This is the reason for the use of monoids instead of the algebraic structure of groups.

We now show how a commutative monoid can be explicitly constructed using what is called a **presentation**.

The first step is to create a **free commutative monoid**. Given a set $\mathbb{W}$, that describes elemental edges, the free commutative monoid on $\mathbb{W}$ is $\mathcal{W} = (\mathbb{W}^*, \|_f)$, where $\mathbb{W}^*$ is the set of all finite multisets of the elements of $\mathbb{W}$, which represents all possible finite parallels of edges. Here, $\|_f$ encodes the multiset sum (free sum) and the element $0_{\mathcal{W}}$ is the empty multiset. Note that the set $\mathbb{W}$ itself does not need to be finite, or even countable.

At this point, the structure is certainly a commutative monoid. However, it is not yet capable of describing an arbitrary one. In particular, it is blind to the possibility of different sets of edges in parallel being equivalent (with regard to the application at hand). For instance, if we are working with resistors in parallel, we would like to be able to encode into the structure the fact that $30\|15 = 20\|20$, from basic circuit theory.

In order to generalize this, the second step of the procedure is to quotient the free commutative monoid $\mathcal{W}$ over a **congruence relation** $\mathcal{R}$. A congruence relation on an algebraic structure is an equivalence relation that is compatible with that structure. In our case, this means that we require $\mathcal{R}$ to be such, that the quotient $\mathcal{M} = \mathcal{W}/\mathcal{R}$ is a commutative monoid. Here, we think of the equivalence relation $\mathcal{R}$ as a function in $\mathbb{W}^* \to \mathbb{M}$ such that its level sets are the corresponding equivalence classes.

In order to satisfy the compatibility condition, we require that if $\mathcal{R}(a_1) = \mathcal{R}(a_2) = A$ and $\mathcal{R}(b_1) = \mathcal{R}(b_2) = B$ then $\mathcal{R}(a_1\|_f b_1) = \mathcal{R}(a_2\|_f b_2) = A\|B$, for any such $a_1, a_2, b_1, b_2 \in \mathbb{W}^*$. That is, for any equivalence classes, we can choose any of its elements as a representative, and when operating them ($\|_f$) the result should be exactly the same, which defines a consistent operation $\|$ on the equivalence classes.

Note that any commutative monoid has a presentation. Given a commutative monoid $\mathcal{M} = (\mathbb{M}, \|)$, we can create the free monoid $\mathcal{W} = (\mathbb{M}^*, \|_f)$. To this end, define the congruence relation $\mathcal{R}\colon \mathbb{M}^* \to \mathbb{M}$ such that for any element $w = w_1\|_f \ldots \|_f w_k$, with $w \in \mathbb{M}^*$ and $w_i \in \mathbb{M}$, $i \in \{1, \ldots, k\}$, we have $\mathcal{R}(w) = w_1\| \ldots \|w_k$. Then, we have that $\mathcal{M} = \mathcal{W}/\mathcal{R}$.

We can also construct our commutative monoid of interest $\mathcal{M}$ using the set that describes the elemental edges $\mathbb{W}$ and defining the congruence relation $\mathcal{R}$ implicitly using a set of equations E. This can be written as $\mathcal{M} = \langle \mathbb{W} | E \rangle$. In the particular case of a free monoid, we write $\mathcal{M} = \langle \mathbb{W} | \rangle$. We illustrate these concepts with the following examples.

Example 2.1.1. *Consider the commutative monoid generated by finite sets of resistors in parallel. In this case, one has* $\mathcal{M} = \langle \mathbb{W} | E \rangle$, *with*

$$\mathbb{W} = \mathbb{R}_0^+ \cup \{\infty\}$$

and

$$E = \begin{cases} w_1\|w_2 = w_1 w_2/(w_1 + w_2) & \forall w_1, w_2 \in \mathbb{R}_0^+ : w_1 + w_2 \neq 0, \\ w_1\|\infty = w_1 & \forall w_1 \in \mathbb{W}, \\ 0\|0 = 0. & \end{cases}$$

This allows us to verify that indeed $30\|15 = 20\|20$. *In particular, those parallels are equivalent to*

an elemental edge of value 10*. For the case of resistors, any set of parallel edges can be simplified into a single edge in* $\mathbb{W}$*. This is not true in general for an arbitrary commutative monoid.*
The identity of this monoid is $0_{\mathcal{M}} = \infty$*. Note that there is no element in* $\mathcal{M}$*, except for the identity* $0_{\mathcal{M}}$ *that has an inverse. That is, if there is a finite resistor w between two nodes, there is no resistor* w^{-1} *that we can add in parallel that will cancel it, that is* $w \| w^{-1} = 0_{\mathcal{M}} = \infty$*.* □

Remark 2.1.2. *Note that this formalism is extremely general. It allows us to parameterize individual edges with anything we might want, such as complex numbers, vectors, matrices, functions or any data structure as abstract as necessary.* □

In Example 2.1.1 it can be seen that the zero-valued resistor, which is **not** the "zero" of the monoid ($0_{\mathcal{M}}$), is an **annihilator**. That is, an element $a \in \mathcal{M}$ such that $w \| a = a$ for all $w \in \mathcal{M}$. Not every monoid has an annihilator, but if it exists, it is unique.

Example 2.1.3. *Consider the commutative monoid* $\mathcal{M} = (\mathbb{N}, \cdot)$*, that is, the positive integers under the usual product, which has* $0_{\mathcal{M}} = 1$*. Define now the* ***free*** *monoid* $\mathcal{N} = (\mathbb{P}^*, \|_f)$*, where* $\mathbb{P}$ *is the set of prime numbers. We conveniently denote the empty multiset in* $\mathbb{P}^*$*, which corresponds to* $0_{\mathcal{N}}$*, by* 1*. Then, the fundamental theorem of arithmetic says that these monoids are two different ways of describing the exact same object. They are called* ***isomorphic****. This means that there is a bijective mapping* $f \colon \mathbb{P}^* \to \mathbb{N}$ *that preserves the monoid structure (isomorphism). In particular,* $f(p_1 \|_f p_2) = f(p_1) \cdot f(p_2)$ *for all* $p_1, p_2 \in (\{1\} \cup \mathbb{P})$ *and* $f(0_{\mathcal{N}}) = 0_{\mathcal{M}}$*. We can find such an* f *by defining* $f\left(\sum_{i=1}^{k} p_i\right) = \prod_{i=1}^{k} p_i$*, in which* $\sum$ *is with regard to the multiset sum* $\|_f$*. This satisfies* $f(1) = 1$ *and the bijectivity comes from the uniqueness of prime factorization.* □

Remark 2.1.4. *Note that for the monoid* $\mathcal{N}$ *in Example 2.1.3, in opposition to the resistor case (Example 2.1.1), two elemental edges in parallel are almost never equivalent to another elemental edge. In fact, the only exception is the parallel with identity elements, for which this is inevitable.*
□

Example 2.1.5. *The set of generalized functions together with the convolution operation forms a commutative monoid. Its identity is* $\delta(\cdot)$*, the Dirac delta distribution.* □

Example 2.1.6. *Consider a network with two types of elemental edges, each with its own commutative monoid structure. For instance,* $\mathcal{M}_1 = (\mathbb{R}, +)$ *and* $\mathcal{M}_2 = (\mathbb{R} \to [-1, 1], \cdot)$*.*
We can merge them into a single commutative monoid by doing a direct product $\mathcal{M} = \mathcal{M}_1 \times \mathcal{M}_2$*.*
An element $m \in \mathcal{M}$ *is an ordered pair* (m_1, m_2) *such that* $m_1 \in \mathbb{R}$ *and* $m_2 \in \mathbb{R} \to [-1, 1]$*.*
The operation $\|$ *of the new monoid is then given by*

$$w \| v = (w_1, w_2) \| (v_1, v_2) = (w_1 + v_1, w_2 \cdot v_2),$$

that is, the concatenation of applying the respective monoid operations to each component. The identity element of the new monoid is $0_{\mathcal{M}} = (0_{\mathcal{M}_1}, 0_{\mathcal{M}_2}) = (0, 1)$*.* □

This approach of constructing a commutative monoid $\mathcal{M}$ by merging smaller monoids that represent different edge-types, allows us to use a single monoid structure to fully describe the possible multiedge, multiedge-type connectivity between two cells.
Note that for each particular pair of cell types $i, j \in T$, we could have different monoid structures, which we denote as $\mathcal{M}_{ij}$, with respect to directed edges from cells of type j into cells of type i.
The connectivity of the network can then be described by a single matrix whose entries are elements of the appropriate monoid.

2.2 Multi-indexes

A multi-index is an ordered n-tuple of non-negative integers (indexes). That is, an element of $\mathbb{N}_0^n$.
Two particularly important multi-indexes are $\mathbf{0}_n$ and $\mathbf{1}_n$, which represent the tuple of n zeros and the tuple of n ones, respectively. Furthermore, we denote by $\mathbf{e}_j$ the tuple such that its j^{th} entry is 1 and all the others are zero.
We denote the multi-indexes with the same notation we use for vectors, using bold, as in $\mathbf{k} = [k_1, \ldots .k_n]^\top$. Their norm is defined as $|\mathbf{k}| := \sum_{i=1}^n k_i$.
The elements (of the same tupleness n) can be multiplied by non-negative integers and added together freely, although subtraction and division are not always well-defined. For instance,

$$\mathbf{k} = 2\,\mathbf{1}_3 + 3\,\mathbf{e}_2 = \begin{bmatrix} 2 \\ 5 \\ 2 \end{bmatrix}.$$

The multi-indexes (of the same tupleness n) form a partial order in the straightforward way, that is, $\mathbf{k^1} \geq \mathbf{k^2}$ if and only if $k_i^1 \geq k_i^2$ for every entry $1 \leq i \leq n$. Note that for $n > 1$ the order is partial since neither $\mathbf{k^1} \geq \mathbf{k^2}$ nor $\mathbf{k^1} \leq \mathbf{k^2}$ are required. This happens when there are $1 \leq i, j \leq n$ such that $k_i^1 > k_i^2$ and $k_j^1 < k_j^2$. In this case we say that the pair $(\mathbf{k}_1, \mathbf{k}_2)$ is non-comparable.
We often specify the tupleness n of a multi-index $\mathbf{k}$ indirectly, by using $\mathbf{k} \geq \mathbf{0}_n$ in order to denote $\mathbf{k} \in \mathbb{N}_0^n$, or $\mathbf{k} \geq \mathbf{1}_n$ to denote $\mathbf{k} \in \mathbb{N}^n$.

2.3 Weighted coupled cell networks

A general weighted coupled cell network is given by the following definition.

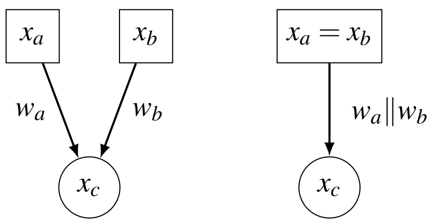
Definition 2.3.1. *A network $\mathcal{G}$ consists of a set of cells $\mathcal{C}_\mathcal{G}$, where each cell has a type, given by a set T according to $\mathcal{T}_\mathcal{G}\colon\ \mathcal{C}_\mathcal{G} \to T$ and has an $|\mathcal{C}_\mathcal{G}| \times |\mathcal{C}_\mathcal{G}|$ in-adjacency matrix $M_\mathcal{G}$. The entries of $M_\mathcal{G}$ are elements of a family of commutative monoids $\{\mathcal{M}_{ij}\}_{i,j \in T}$ such that $[M_\mathcal{G}]_{cd} = m_{cd} \in \mathcal{M}_{ij}$, for any cells $c, d \in \mathcal{C}_\mathcal{G}$ with types $i = \mathcal{T}_\mathcal{G}(c)$, $j = \mathcal{T}_\mathcal{G}(d)$.* □

We often have the set of cell types be of the form $T = \{1, \ldots, |T|\}$ so that it is simple to index. For each commutative monoid $\mathcal{M}_{ij}$ we denote its "zero" element as 0_{ij}. The entries of m_{cd} are able to encode the complete connectivity (multiedge, multiedge-type) of the directed edges from d to c thanks to the algebraic structure of the commutative monoid.

Remark 2.3.2. *The subscripts $_\mathcal{G}$ are omitted when the network of interest is clear from context.* □

2.4 Admissibility

In this section, we describe the minimal properties that we require for a function $f\colon \mathbb{X} \to \mathbb{Y}$ to satisfy in order to be a plausible modeling of the dynamics $\dot{\mathbf{x}}/\mathbf{x}^+ = f(\mathbf{x})$ or some measurement function $\mathbf{y} = f(\mathbf{x})$ on the network. We call such a function **admissible** on the network of interest. In particular, such a function describes some **first-order property** of the network. That is, it models something that, when evaluated at cell, depends on the state of that cell and its in-neighbors. This does not mean that everything on a network has to (or can) be defined by such a function. For instance, the second derivative or the two-step evolution of a dynamical system on a network will not be of this form. Those functions will be "second-order" in the sense that, when evaluated on a cell, they depend on the states of that cell, together with the states of the cells in its first and second in-neighborhoods (neighbor of neighbor). Such second-order functions are, however, fully defined from the original first-order functions.
We construct admissible functions through the use of mathematical objects called **oracle components**, first introduced in Sequeira et al. (2021) and then simplified in Sequeira et al. (2023a). An oracle component is a mathematical object that describes how cells of a given type respond to arbitrary finite in-neighborhoods. It completely separates the modeling of the behavior of cells from the particular network on which the cells of interest are inserted.
Consider the simple network of Figure 2.5a, (which could be part of a larger network) consisting of cell c and its in-neighborhood. We have cell types $T = \{1,2\}$ which represent "circle" and



(a) Original. (b) Merged.

Figure 2.5: Edge merging.

"square" cells, respectively. In order to define functions on the cells we associate with them the state sets $\mathbb{X}_1, \mathbb{X}_2$ and the output sets $\mathbb{Y}_1, \mathbb{Y}_2$ according to their respective type.
We consider that the input received by a cell is independent of how we draw the network, that is, from the point of view of cell c, there would be no difference if cell b was at the left of cell a.

Then, for a function $\hat{f}_1$ acting on cells of type 1, we would expect that

$$\hat{f}_1\left(x_c; \begin{bmatrix} w_a \\ w_b \end{bmatrix}, \begin{bmatrix} x_a \\ x_b \end{bmatrix}\right) = \hat{f}_1\left(x_c; \begin{bmatrix} w_b \\ w_a \end{bmatrix}, \begin{bmatrix} x_b \\ x_a \end{bmatrix}\right),$$

for $x_c \in \mathbb{X}_1$, $x_a, x_b \in \mathbb{X}_2$ and $w_a, w_b \in \mathcal{M}_{12}$. Moreover, since cells a and b are of the same cell type (square) ($\mathcal{T}(a) = \mathcal{T}(b) = 2$), we expect that when they are in the same state ($x_a = x_b = x_{ab}$), the total input received by cell c at that instant, is the same as if both edges originated from a single "square" cell with that state, as in Figure 2.5b. That is,

$$\hat{f}_1\left(x_c; \begin{bmatrix} w_a \\ w_b \end{bmatrix}, \begin{bmatrix} x_{ab} \\ x_{ab} \end{bmatrix}\right) = \hat{f}_1\left(x_c; w_a \| w_b, x_{ab}\right).$$

Although this might look inconsistent since the domains look mismatched, the following definition formalizes it in a rigorous way. Finally, when $\hat{f}_1$ is evaluated at a cell it should only depend on the in-neighborhood of that cell. Therefore, if $w_a = 0_{12}$, cell c should not be directly influenced by cell a. That is,

$$\hat{f}_1\left(x_c; \begin{bmatrix} 0_{12} \\ w_b \end{bmatrix}, \begin{bmatrix} x_a \\ x_b \end{bmatrix}\right) = \hat{f}_1\left(x_c; w_b, x_b\right).$$

These ideas are now formalized in the following definition.

Definition 2.4.1. *Consider a given set of cell types T, and some related sets $\{\mathbb{X}_j, \mathbb{Y}_j\}_{j\in T}$ together with a family of commutative monoids $\{\mathcal{M}_{ij}\}_{j\in T}$, for a given fixed $i \in T$. Take a function $\hat{f}_i$ defined on*

$$\hat{f}_i \colon \mathbb{X}_i \times \mathop{\mathring{\bigcup}}_{\mathbf{k}\geq \mathbf{0}_{|T|}} \left(\mathcal{M}_i^{\mathbf{k}} \times \mathbb{X}^{\mathbf{k}}\right) \to \mathbb{Y}_i, \tag{2.1}$$

where $\mathring{\bigcup}$ denotes the disjoint union and for multi-index $\mathbf{k}$ we define $\mathbb{X}^{\mathbf{k}} := \mathbb{X}_1^{k_1} \times \ldots \times \mathbb{X}_{|T|}^{k_{|T|}}$ and $\mathcal{M}_i^{\mathbf{k}} := \mathcal{M}_{i1}^{k_1} \times \ldots \times \mathcal{M}_{i|T|}^{k_{|T|}}$.
*The function $\hat{f}_i$ is called an **oracle component of type i**, if it has the following properties:*

1. *If σ is a **permutation** matrix (of appropriate dimension), then*

$$\hat{f}_i(x; \mathbf{w}, \mathbf{x}) = \hat{f}_i(x; \sigma\mathbf{w}, \sigma\mathbf{x}), \tag{2.2}$$

where we assume, without loss of generality, that one can keep track of the cell types of each element of $\sigma\mathbf{w}$ and $\sigma\mathbf{x}$.

2. *If the indexes j_1, j_2 and j_{12} denote cells of type $j \in T$, then*

$$\hat{f}_i\left(x; \begin{bmatrix} w_{j_1} \| w_{j_2} \\ \mathbf{w} \end{bmatrix}, \begin{bmatrix} x_{j_{12}} \\ \mathbf{x} \end{bmatrix}\right) = \hat{f}_i\left(x; \begin{bmatrix} w_{j_1} \\ w_{j_2} \\ \mathbf{w} \end{bmatrix}, \begin{bmatrix} x_{j_{12}} \\ x_{j_{12}} \\ \mathbf{x} \end{bmatrix}\right). \tag{2.3}$$

3. *If the index j denotes a cell of type $j \in T$, then*

$$\hat{f}_i\left(x; \begin{bmatrix} 0_{ij} \\ \mathbf{w} \end{bmatrix}, \begin{bmatrix} x_j \\ \mathbf{x} \end{bmatrix}\right) = \hat{f}_i\left(x; \mathbf{w}, \mathbf{x}\right). \tag{2.4}$$

□

The disjoint union allows us to distinguish neighborhoods of different types. That is, even in the particular case of $\mathbb{X}_1 = \mathbb{X}_2$ and $\mathcal{M}_{i1} = \mathcal{M}_{i2}$, we are able to differentiate the part of the domain associated with $\mathcal{M}_{i1}^2 \times \mathbb{X}_1^2$ from the one associated with $\mathcal{M}_{i1} \times \mathcal{M}_{i2} \times \mathbb{X}_1 \times \mathbb{X}_2$. A non-disjoint union, on the other hand, would merge these sets together.

Remark 2.4.2. *As stated in item 1 of Definition 2.4.1, it is always assumed that given any weight w_c or state x_c, we always know the cell type of the corresponding cell c. Note that one can always do enough bookkeeping in order to ensure this. For instance, one can extend $\hat{f}_i(x; \mathbf{w}, \mathbf{x})$ into $\hat{f}_i(x; \mathbf{t}, \mathbf{w}, \mathbf{x})$, where $\mathbf{t}$ would be a vector that encodes the cell types associated with $\mathbf{w}, \mathbf{x}$. Then, we would have $\hat{f}_i(x; \mathbf{t}, \mathbf{w}, \mathbf{x}) = \hat{f}_i(x; \sigma\mathbf{t}, \sigma\mathbf{w}, \sigma\mathbf{x})$ instead.*
Our implicit bookkeeping means that we do not have to constrain σ to preserve cell typing. That is, if we assume some canonical order of the cell types in the part of the domain $\mathcal{M}_i^{\mathbf{k}} \times \mathbb{X}^{\mathbf{k}}$ in Equation (2.1), then we know the correct $\mathbf{k} \geq \mathbf{0}_{|T|}$ and can reorder the rows of $\mathbf{w}$ and $\mathbf{x}$ in $\hat{f}_i(x; \mathbf{w}, \mathbf{x})$ appropriately.
Note that by considering invariance under general permutations, and not having to worry about preserving cell types or respecting some canonical ordering of cell types, we are always able to shift the cells of major interest to the top of the vectors, as in Equations (2.3) and (2.4), regardless of the types of other cells. □

We consider the function $\mathcal{K}$ such that for a set of cells $\mathbf{s}$, we have that $\mathbf{k} = \mathcal{K}(\mathbf{s})$ is the $|T|$-tuple such that k_i is the number of cells in $\mathbf{s}$ that are of type $i \in T$. This allows us to pick the proper $\mathbf{k} \geq \mathbf{0}_{|T|}$ in Equation (2.1) when we want to evaluate oracle components at a cell and its in-neighbors.
The **oracle set** is the set of all $|T|$-tuples of oracle components, such that each element of the tuple represents one of the types in T. It is denoted as

$$\hat{\mathcal{F}}_T = \prod_{i \in T} \hat{\mathcal{F}}_i,$$

where $\hat{\mathcal{F}}_i$ is the set of all oracle components of type i. We are always implicitly assuming sets $\{\mathbb{X}_i, \mathbb{Y}_i\}_{i \in T}$ and commutative monoids $\{\mathcal{M}_{ij}\}_{i,j \in T}$. Note that modeling some aspect of a network that follows our assumptions is effectively choosing one of the elements of $\hat{\mathcal{F}}_T$, which we call **oracle functions**.

Example 2.4.3. *Consider again Example 2.0.1, where we present in Figure 2.3 an instance of a discrete-time cell. This particular cell has three bits coming from edges w_{and} and four bits coming from edges w_{or}, which means that its associated dynamics are given by*

$$x_{n+1} = xor(and(b_1,b_2,b_3), or(b_4,b_5,b_6,b_7)).$$

In general, we can write the dynamics of a cell of this type for the case where we have arbitrary (finite) inputs of each edge type $\{w_{and}, w_{or}\}$. That is, the corresponding oracle component. In particular, the dynamics associated with this cell is, in the general case, given by

$$x_{n+1} = \begin{cases} 1 \quad if \begin{cases} (\#(w_{and},0) = 0 \text{ and } \#(w_{or},1) = 0) \\ or \\ (\#(w_{and},0) > 0 \text{ and } \#(w_{or},1) > 0), \end{cases} \\ 0 \quad otherwise, \end{cases}$$

where $\#(w,b)$ denotes the cardinality of edges with weight w that have the bit b as input. □

Example 2.4.4. *Consider again Example 2.0.2, where we present in Figure 2.4 an instance of a continuous-time network with two cells. The oracle component corresponding to cells of this type is such that for an arbitrary cell c on a network with this type of cells, with neighbors $\mathbf{s}$, we have that*

$$\dot{x}_c = \hat{f}_i(x_c; \mathbf{w_s}, \mathbf{x_s}) = \frac{1}{C} \sum_{d \in \mathbf{s}} \frac{1}{w_d}(x_d - x_c).$$

Note that the dynamics of this particular network, which are given by

$$\dot{x}_a = \frac{1}{C} \frac{1}{R_1 \| R_2}(x_b - x_a),$$
$$\dot{x}_b = \frac{1}{C} \frac{1}{R_1 \| R_2}(x_a - x_b),$$

are directly obtained by evaluating the oracle component to each cell of the network according to its particular in-neighborhood. □

Definition 2.4.5. *Consider a network $\mathcal{G}$ on a cell set $\mathcal{C}$ with cell types in T according to the cell type partition $\mathcal{T}$, and an in-adjacency matrix M. Assume without loss of generality that the cells are ordered according to the cell types such that we can associate with the network a state $\mathbb{X} := \mathbb{X}^{\mathbf{k}}$ and output $\mathbb{Y} := \mathbb{Y}^{\mathbf{k}}$ sets, with $\mathbf{k} = \mathcal{K}(\mathcal{C})$.*
A function $f\colon \mathbb{X} \to \mathbb{Y}$, given as

$$f = (f_c)_{c \in \mathcal{C}}, \quad with\ f_c\colon \mathbb{X} \to \mathbb{Y}_i, \quad i = \mathcal{T}(c),$$

is said to be $\mathcal{G}$***-admissible*** *if there is some oracle function* $\hat{f} \in \hat{\mathcal{F}}_T$, $\hat{f} = (\hat{f}_i)_{i\in T}$ *such that*

$$f_c(\mathbf{x}) = \hat{f}_i\left(x_c; \mathbf{m}_c^\top, \mathbf{x}\right), \tag{2.5}$$

for $\mathbf{x} \in \mathbb{X}$*, where* x_c *is the* c^{th} *coordinate of* $\mathbf{x}$ *and* $\mathbf{m}_c$ *is the* c^{th} *row of matrix* M*. In this case we write* $f = \hat{f}|_{\mathcal{G}}$. □

The set of all $\mathcal{G}$-admissible functions is denoted as $\mathcal{F}_{\mathcal{G}}$. It can be thought of as the result of evaluating $\hat{\mathcal{F}}_T$ at $\mathcal{G}$, which can be written as $\hat{\mathcal{F}}_T|_{\mathcal{G}}$. Note that process of evaluating oracle functions at a network is not necessarily injective. There might be oracle functions $\hat{f}, \hat{g} \in \hat{\mathcal{F}}_T$ with $\hat{f} \neq \hat{g}$ such that $\hat{f}|_{\mathcal{G}} = \hat{g}|_{\mathcal{G}}$.
The next example makes explicit the relation between the connectivity graph of a network and how that constrains any possible admissible function that acts on it.

Example 2.4.6. *Figure 2.6 shows an example of a* ***CCN*** *of three cells. We have cell types* $T = \{1, 2\}$ *which represent "circle" and "square" cells, respectively. This* ***CCN*** *can be described by*

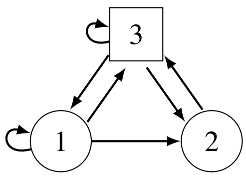


Figure 2.6: Simple network with admissible functions that have the structure given by Equations (2.7) to (2.9).

the in-adjacency matrix M

$$M = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \tag{2.6}$$

together with the cell type partition $\mathcal{T} = \{\{1, 2\}, \{3\}\}$*. This means that a suitable* $f \in \mathcal{F}_{\mathcal{G}}$ *should have the following structure*

$$f_1(\mathbf{x}) = \hat{f}_1(x_1; \begin{bmatrix} 1 & 0 & 1 \end{bmatrix}^\top, \mathbf{x}), \tag{2.7}$$

$$f_2(\mathbf{x}) = \hat{f}_1(x_2; \begin{bmatrix} 1 & 0 & 1 \end{bmatrix}^\top, \mathbf{x}), \tag{2.8}$$

$$f_3(\mathbf{x}) = \hat{f}_2(x_3; \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^\top, \mathbf{x}), \tag{2.9}$$

for some $\hat{f} \in \hat{\mathcal{F}}_T$. □

To make more explicit the importance of a rigorous definition for admissibility, the following example presents a case that might look reasonable at a first glance but ends up not being admissible.

Example 2.4.7. *Consider the simple network in Figure 2.5 that was used to illustrate the edge merging concept. We will propose a function on the original network Figure 2.5a and verify if it satisfies our assumptions.*
We consider that the cells have associated state and output sets given by $\mathbb{X}_1 = \mathbb{X}_2 = \mathbb{Y}_1 = \mathbb{R}$, such that $T = \{1,2\}$ identify the cell types "circle" and "square" respectively.
The directed edges from "square" into "circle" are in $\mathcal{M}_{12}$. Given functions $g\colon \mathbb{R} \to \mathbb{R}$ and $p\colon \mathcal{M}_{12} \to \mathbb{R}$, with $p(0_{12}) = 0$, it is tempting to think that a function f_c, could be modeled by

$$f_c(\mathbf{x}) = g(x_c) + p(w_a)x_a + p(w_b)x_b + p(w_a)p(w_b)x_a x_b. \tag{2.10}$$

After all, if we simultaneously switch $w_a \leftrightarrow w_b$ and $x_a \leftrightarrow x_b$, f_c would still look the same. Consider, $w_a = w$, $x_a = x$ and $w_b = 0_{12}$. Then, if cell c only had one neighbor (of type square), f_c would be given by

$$f_c(\mathbf{x}) = g(x_3) + p(w)x.$$

If we have $x_a = x_b = x_{ab}$, from the edge-merging principle, we should be in the situation of Figure 2.5b. We would have

$$f_c(\mathbf{x}) = g(x_c) + p(w_a \| w_b)x_{ab}.$$

However, from direct substitution on Equation (2.10) we obtain

$$f_c(\mathbf{x}) = g(x_c) + (p(w_a) + p(w_b) + p(w_a)p(w_b)x_{ab})x_{ab},$$

which means that this is not admissible, except for the trivial case $p = 0$, since

$$p(w_a \| w_b) = p(w_a) + p(w_b) + p(w_a)p(w_b)x_{ab}$$

goes against the assumption that p depends only on the edge weights.
Consider that f_c was modeled instead as

$$f_c(\mathbf{x}) = g(x_c) + p(w_a)x_a + p(w_b)x_b + p(w_a)p(w_b)\frac{x_a + x_b}{2}. \tag{2.11}$$

Following the exact same approach this requires

$$p(w_a \| w_b) = p(w_a) + p(w_b) + p(w_a)p(w_b), \tag{2.12}$$

which is a valid constraint. It only depends on its inputs and is compatible with a commutative *monoid structure, that is,*

$$p(w\|0_{12}) = p(w),$$
$$p(w_1\|w_2) = p(w_2\|w_1),$$
$$p((w_1\|w_2)\|w_3) = p(w_1\|(w_2\|w_3)).$$

Note that for each of the three equalities, the inputs for both members are the same element of $\mathcal{M}_{12}$. The same input of a function has to output the same value.
Note that Equation (2.12) is only a necessary condition, not a sufficient one. In order to be systematic, we need to show that there is one underlying oracle component $\hat{f}_1$ that models how cells of type "circle" behave under arbitrary neighborhoods. Conjecture the candidate oracle component

$$\hat{f}_1\left(x;\mathbf{w_s},\mathbf{x_s}\right) = g(x) + \sum_{c\in\mathbf{s}} p(w_c)x_c + \sum_{\substack{\{c,d\}\subseteq\mathbf{s}\\ c\neq d}} p(w_c)p(w_d)\frac{x_c+x_d}{2}, \tag{2.13}$$

dependent only on its "square" neighbors and itself. Note that, when evaluated at a neighborhood of two "square" cells, this maps into Equation (2.11). We need to verify if Equation (2.13) is a valid oracle component. Items 1 and 3 of Definition 2.4.1 are immediate to verify (with $p(0_{12}) = 0$). We now verify item 2 of Definition 2.4.1, that is

$$\hat{f}_i\left(x;\begin{bmatrix} w_{j_1}\|w_{j_2} \\ \mathbf{w_s}\end{bmatrix},\begin{bmatrix} x_{j_{12}} \\ \mathbf{x_s}\end{bmatrix}\right) = \hat{f}_i\left(x;\begin{bmatrix} w_{j_1} \\ w_{j_2} \\ \mathbf{w_s}\end{bmatrix},\begin{bmatrix} x_{j_{12}} \\ x_{j_{12}} \\ \mathbf{x_s}\end{bmatrix}\right).$$

Expanding according to Equation (2.13) and canceling common terms get us that

$$p(w_{j_1}\|w_{j_2})x_{j_{12}} + \sum_{c\in\mathbf{s}} p(w_{j_1}\|w_{j_2})p(w_c)\frac{x_{j_{12}}+x_c}{2}$$

is equal to

$$\left[p(w_{j_1}) + p(w_{j_2}) + p(w_{j_1})p(w_{j_2})\right]x_{j_{12}} + \sum_{c\in\mathbf{s}}\left[p(w_{j_1}) + p(w_{j_2})\right]p(w_c)\frac{x_{j_{12}}+x_c}{2}.$$

Using the constrain in Equation (2.12) that we found previously, this means that

$$p(w_{j_1})p(w_{j_2})\sum_{c\in\mathbf{s}} p(w_c)\frac{x_{j_{12}}+x_c}{2} = 0.$$

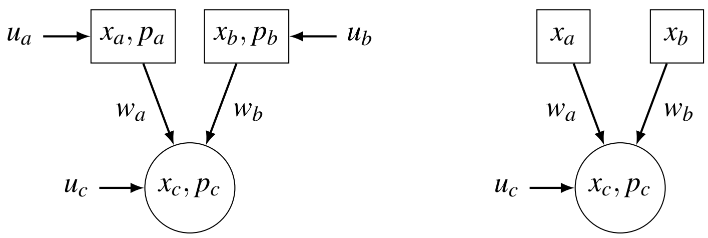
This is only satisfied for every possible combination of weights and states if we are again in the trivial case $p = 0$.
This underlines the importance of the concept of oracle components. Even though Equation (2.11) looks suitable when we test it in neighborhoods of zero, one and two cells, which might even be the only ones present in our particular network of interest, there has to exist some underlying function

that describes how the cell interacts under other types of neighborhoods. Since the behavior of a cell under different neighborhoods is not independent, even neighborhoods that we do not care about impose constraints on the ones that we do care about.
We verified that Equation (2.13) does not define (for a non-trivial p) a valid oracle component. This does not prove yet that Equation (2.11) is impossible, there might still be some more complicated oracle component that maps into Equation (2.11).
*In summary, it is essential to define a valid oracle function in order to model an admissible function on a network. It is **not** enough to define admissible functions for the neighborhoods we are interested in and verifying if those are cross-compatible.* □

At this point it is still not clear what exactly are our degrees of freedom in order to construct a valid oracle component. This is explored in Sections 5.2 and 5.3.

2.5 Extension for exogenous inputs and inner cell parameters

This formalism can easily be generalized so that it allows for inner cell parameters and can deal with **exogenous** inputs on the cells. By exogenous we mean that they are external from the point of view of the network. On the other hand, we think of the influence that a neighbor of a cell has on it as **endogenous**.
Consider the simple network in Figure 2.7a, where we illustrate a cell c and its in-neighborhood. We denote on each cell their inner cell parameters p_a, p_b, p_c and their exogenous inputs u_a, u_b, u_c. We assume that the inner cell parameters p_a, p_b and the exogenous inputs u_a, u_b of cells a, b do not affect cell c directly. Instead, they affect cell c indirectly by affecting the state evolution of x_a, x_b on which c depends. In summary, we assume that at any given time, the cell c only "sees" the part of the network illustrated in Figure 2.7b. This means, that we adapt the concept of oracle



(a) Input set of cell c.

(b) Equivalent neighborhood from the point of view of cell c.

Figure 2.7: Network with inner cell parameters and exogenous inputs.

functions such that a suitable admissible function on c, has the following structure

$$f_c(\mathbf{x}) = \hat{f}_1\left(x_c, p_c, u_c; \begin{bmatrix} w_a \\ w_b \end{bmatrix}, \begin{bmatrix} x_a \\ x_b \end{bmatrix}\right), \tag{2.14}$$

for some $\hat{f} \in \hat{\mathcal{F}}_T$.
We can extend Definition 2.4.1 such that oracle components are instead defined on

$$\hat{f}_i \colon \mathbb{X}_i \times \mathbb{P}_i \times \mathbb{U}_i \times \overset{\circ}{\bigcup_{\mathbf{k} \geq \mathbf{0}_{|T|}}} \left(\mathcal{M}_i^{\mathbf{k}} \times \mathbb{X}^{\mathbf{k}}\right) \to \mathbb{Y}_i, \tag{2.15}$$

where $\mathbb{P}_i$ is the set of possible inner parameters of a cell and $\mathbb{U}_i$ denotes the set of possible exogenous inputs. Since we assume these characteristics to only affect directly the cell it refers to, the extension is trivial. All the basic properties and definitions are basically unchanged. The only things that change are the parts before the semicolon, which change from $\hat{f}_i(x; \mathbf{w}, \mathbf{x})$ into $\hat{f}_i(x, p, u; \mathbf{w}, \mathbf{x})$.

Chapter 3

Equality-based synchronism

In this chapter, we concern ourselves with patterns of synchronism defined by equalities between the states of cells. Such a set of equalities establishes an equivalence relation, which we encode through the use of partitions on the set of cells.

We generalize the known results in Stewart (2007) regarding lattices of balanced partitions $\Lambda_{\mathcal{G}}$ for the case where we are interested about invariance under some subset of admissible functions $F \subseteq \mathcal{F}_{\mathcal{G}}$. In particular, we show that the set of partitions L_F that are invariant under F always form lattices. Furthermore, these lattices share with $\Lambda_{\mathcal{G}}$ the very special properties of always containing the trivial partition $(\bot)$ and being closed under the standard partition join $(\vee)$. For general lattice theory refer to Davey and Priestley (2002).

We study with some detail the very particular case of lattices L such that $\vee_L = \vee$ and $\bot_L = \bot$ and then show that all the lattices regarding invariant synchrony patterns are of this type. Furthermore, such lattices always have an associated cir_L function. We show that the know results regarding balanced partitions and quotient networks Stewart et al. (2003); Golubitsky et al. (2005); Golubitsky and Stewart (2006) generalize to the weighted framework. Here, $\Lambda_{\mathcal{G}}$ being closed under the join $\vee$ is proved in a novel, algebraic way, instead of the usual duality argument between balanced partitions and invariant subspaces. Finally, we improve the **CIR** algorithm for finding balanced partitions in a manner that works for the weighted case.

3.1 Partitions and their representations

A partition $\mathcal{A}$ on a set of cells $\mathcal{C}$ is a set of non-empty subsets of $\mathcal{C}$ such that they are pairwise disjoint and their union is equal to $\mathcal{C}$. We often refer to each element of a given partition (corresponding to a subset of cells) by the term **color**. The number of colors in a partition is called its **rank**.

We construct the **quotient set** $\mathcal{C}/\mathcal{A}$ by taking the elements of $\mathcal{C}$ and merging them together according to $\mathcal{A}$, such that each color of $\mathcal{A}$ is associated with an element of $\mathcal{C}/\mathcal{A}$. We can now think of $\mathcal{A}$ as a function from $\mathcal{C}$ to $\mathcal{C}/\mathcal{A}$, which we illustrate in the following example.

Example 3.1.1. *Consider the set of cells $\mathcal{C} = \{a,b,c,d,e\}$. Then, $\mathcal{A} = \{\{a,b\},\{c\},\{d,e\}\}$ is a partition on $\mathcal{C}$ with three colors ($rank(\mathcal{A}) = 3$). We denote the quotient set as $\mathcal{C}/\mathcal{A} = \{ab,c,de\}$, which contains three elements. Then, $\mathcal{A}$ acts as function in $\mathcal{C} \to \mathcal{C}/\mathcal{A}$, and we write $\mathcal{A}(a) = \mathcal{A}(b) = ab$, $\mathcal{A}(c) = c$ and $\mathcal{A}(d) = \mathcal{A}(e) = de$.* □

Remark 3.1.2. *In the example above it might look more canonical to think of the elements of $\mathcal{C}/\mathcal{A}$ as $ab := \{a,b\}$, $c := \{c\}$ and $de := \{d,e\}$. That is, each of its elements is a color according to partition $\mathcal{A}$. However, using this notation, $\mathcal{A}$ and $\mathcal{C}/\mathcal{A}$ would look indistinguishable. We want to think of these objects as semantically different. While we think of a partition $\mathcal{A}$ as a set of sets of elements (cells), we think of $\mathcal{C}/\mathcal{A}$ as just a set of elements, (which are colors, and therefore end up being sets themselves). In order to make this clear we use this shorthand notation. This becomes more important when we compose partitions (e.g., we apply a partition on the set $\mathcal{C}/\mathcal{A}$) and define the concept of partition quotients.* □

Interpreting partitions as functions allows us to say that two cells $c,d \in \mathcal{C}$ are of the same color, according to $\mathcal{A}$, if and only if $\mathcal{A}(c) = \mathcal{A}(d)$. Furthermore, they are surjective functions by construction and each color is given by the preimage of each element of $\mathcal{C}/\mathcal{A}$. Conversely, note that every surjective function establishes a partition on its domain through its level sets.
Given two partitions $\mathcal{A}$, $\mathcal{B}$ on a set of cells $\mathcal{C}$, we say that $\mathcal{A}$ is **finer** than $\mathcal{B}$, denoted as $\mathcal{A} \leq \mathcal{B}$, if

$$\mathcal{A}(c) = \mathcal{A}(d) \implies \mathcal{B}(c) = \mathcal{B}(d) \tag{3.1}$$

for all $c,d \in \mathcal{C}$. Conversely, $\mathcal{B}$ is said to be **coarser** than $\mathcal{A}$. Roughly speaking, Equation (3.1) means that if any pair of cells have the same color according to partition $\mathcal{A}$, then they also have the same color according to $\mathcal{B}$. In other words, if we merge some of the colors of $\mathcal{A}$ together, we can obtain $\mathcal{B}$. Conversely, we can obtain $\mathcal{A}$ by starting with $\mathcal{B}$ and splitting some of its colors into smaller ones. The **trivial** partition, in which each color consists of a single cell, is the **finest** and its rank is $|\mathcal{C}|$. We now show that if $\mathcal{A} \leq \mathcal{B}$ we can define a **quotient partition** $\mathcal{B}/\mathcal{A}$.

Lemma 3.1.3. *Consider a set of cells $\mathcal{C}$ and the partitions $\mathcal{A} : \mathcal{C} \to \mathcal{C}/\mathcal{A}$ and $\mathcal{B} : \mathcal{C} \to \mathcal{C}/\mathcal{B}$. Then, $\mathcal{A} \leq \mathcal{B}$ if and only if there is some $\mathcal{B}/\mathcal{A} : \mathcal{C}/\mathcal{A} \to \mathcal{C}/\mathcal{B}$ such that $\mathcal{B}/\mathcal{A} \circ \mathcal{A} = \mathcal{B}$.* □

Proof. Note that $\mathcal{B}/\mathcal{A} \circ \mathcal{A} = \mathcal{B}$ means that $\mathcal{B}/\mathcal{A}(\mathcal{A}(c)) = \mathcal{B}(c)$ for all $c \in \mathcal{C}$. This means that $\mathcal{B}/\mathcal{A}$ is the function that maps $\mathcal{A}(c) \in \mathcal{C}/\mathcal{A}$ into $\mathcal{B}(c) \in \mathcal{C}/\mathcal{B}$ for all $c \in \mathcal{C}$. Note that this is enough to define $\mathcal{B}/\mathcal{A}$ on its whole domain since $\mathcal{A}$ is surjective. That is, for every element $k \in \mathcal{C}/\mathcal{A}$ there is some $c \in \mathcal{C}$ such that $\mathcal{A}(c) = k$. Finally, $\mathcal{B}/\mathcal{A}$ exists if and only if such a function is well-defined. That is, for every $k \in \mathcal{C}/\mathcal{A}$, the mapping of $k = \mathcal{A}(c)$ into $\mathcal{B}(c)$ has to be completely independent of the particular choice of $c \in \mathcal{C}$, which is equivalent to $\mathcal{A} \leq \mathcal{B}$. ■

In particular, if $\mathcal{A} \leq \mathcal{B}$, the partition $\mathcal{B}/\mathcal{A}$ describes how to merge the colors of $\mathcal{A}$ into the colors of $\mathcal{B}$. Furthermore, note that $\mathcal{B}/\mathcal{A}$ is uniquely defined and is also surjective. If we consider the particular case $\mathcal{B} = \mathcal{A}$, then we have that $\mathcal{A}/\mathcal{A} : \mathcal{C}/\mathcal{A} \to \mathcal{C}/\mathcal{A}$ is such that $\mathcal{A}/\mathcal{A} \circ \mathcal{A} = \mathcal{A}$. That is, $\mathcal{A}/\mathcal{A}$ acts as the identity map in the set $\mathcal{C}/\mathcal{A}$ and it is the trivial partition in that set.

Example 3.1.4. *Consider the set of cells* $\mathcal{C} = \{a,b,c,d,e\}$ *on which we define the partitions* $\mathcal{A} = \{\{a,b\},\{c\},\{d,e\}\}$ $\mathcal{B} = \{\{a,b\},\{c,d,e\}\}$. *We denote the quotient sets as* $\mathcal{C}/\mathcal{A} = \{ab,c,de\}$ *and* $\mathcal{C}/\mathcal{B} = \{ab,cde\}$. *Consider that the mappings* $\mathcal{A}:\mathcal{C}\to\mathcal{C}/\mathcal{A}$ *and* $\mathcal{B}:\mathcal{C}\to\mathcal{C}/\mathcal{B}$ *are defined in the expected way. Then, since we have* $\mathcal{A}\leq\mathcal{B}$, *the quotient partition* $\mathcal{B}/\mathcal{A}:\mathcal{C}/\mathcal{A}\to\mathcal{C}/\mathcal{B}$ *is such that* $\mathcal{B}/\mathcal{A}(ab) = ab$ *and* $\mathcal{B}/\mathcal{A}(c) = \mathcal{B}/\mathcal{A}(de) = cde$.
Using the set of colors notation, we can write $\mathcal{B}/\mathcal{A} = \{\{ab\},\{c,de\}\}$. *Finally, note that* $rank(\mathcal{B}/\mathcal{A}) = rank(\mathcal{B}) = 2$. □

It should be clear that $rank(\mathcal{B}/\mathcal{A}) = rank(\mathcal{B})$ is true in general, since it always corresponds to the size of their common image set $\mathcal{C}/\mathcal{B}$.

Lemma 3.1.5. *The partition quotient preserves the partial order relation* $\leq$. *That is, for all partitions* $\mathcal{A},\mathcal{B}_1,\mathcal{B}_2$ *on* $\mathcal{C}$ *such that* $\mathcal{A}\leq\mathcal{B}_1,\mathcal{B}_2$, *we have that* $\mathcal{B}_1\leq\mathcal{B}_2$ *if and only if* $\mathcal{B}_1/\mathcal{A}\leq\mathcal{B}_2/\mathcal{A}$.
□

Proof. Firstly, note that $\mathcal{B}_1/\mathcal{A}$ and $\mathcal{B}_2/\mathcal{A}$ are both partitions on the set $\mathcal{C}/\mathcal{A}$, therefore the statement $\mathcal{B}_1/\mathcal{A}\leq\mathcal{B}_2/\mathcal{A}$ is meaningful.
Since we have that $\mathcal{A}\leq\mathcal{B}_1,\mathcal{B}_2$ from assumption, we can, using Lemma 3.1.3, write $\mathcal{B}_1\leq\mathcal{B}_2$ as $\mathcal{B}_1/\mathcal{A}(\mathcal{A}(c)) = \mathcal{B}_1/\mathcal{A}(\mathcal{A}(d)) \implies \mathcal{B}_2/\mathcal{A}(\mathcal{A}(c)) = \mathcal{B}_2/\mathcal{A}(\mathcal{A}(d))$ for all $c,d\in\mathcal{C}$.
We have to show that this is equivalent to $\mathcal{B}_1/\mathcal{A}(k) = \mathcal{B}_1/\mathcal{A}(l) \implies \mathcal{B}_2/\mathcal{A}(k) = \mathcal{B}_2/\mathcal{A}(l)$ for all $k,l\in\mathcal{C}/\mathcal{A}$. The forward direction comes from the fact that $\mathcal{A}$ is surjective. That is, for all $k,l\in\mathcal{C}/\mathcal{A}$ there are some $c,d\in\mathcal{C}$ such that $\mathcal{A}(c) = k$ and $\mathcal{A}(d) = l$. The backwards direction is immediate from the fact that for all $c,d\in\mathcal{C}$, we have that $\mathcal{A}(c),\mathcal{A}(d)\in\mathcal{C}/\mathcal{A}$. ■

Lemma 3.1.6. *Consider partitions* $\mathcal{A},\mathcal{B}_1,\mathcal{B}_2$ *on* $\mathcal{C}$ *such that* $\mathcal{A}\leq\mathcal{B}_1\leq\mathcal{B}_2$. *Then,* $(\mathcal{B}_2/\mathcal{A})/(\mathcal{B}_1/\mathcal{A}) = \mathcal{B}_2/\mathcal{B}_1$. □

Proof. From Lemma 3.1.5, it is clear that $\mathcal{B}_1/\mathcal{A}\leq\mathcal{B}_2/\mathcal{A}$. Therefore, $(\mathcal{B}_2/\mathcal{A})/(\mathcal{B}_1/\mathcal{A})$ is well-defined.
Furthermore, from Lemma 3.1.3, it can easily been seen that $(\mathcal{B}_2/\mathcal{A})/(\mathcal{B}_1/\mathcal{A})$ and $\mathcal{B}_2/\mathcal{B}_1$ are both mappings in $\mathcal{C}/\mathcal{B}_1\to\mathcal{C}/\mathcal{B}_2$. We now show that

$$(\mathcal{B}_2/\mathcal{A})/(\mathcal{B}_1/\mathcal{A})(k) = \mathcal{B}_2/\mathcal{B}_1(k)$$

for all $k\in\mathcal{C}/\mathcal{B}_1$. Note that $k\in\mathcal{C}/\mathcal{B}_1$ if and only if there is some $c\in\mathcal{C}$ such that $\mathcal{B}_1(c) = k$. Since $\mathcal{A}\leq\mathcal{B}_1$, we have that $k = \mathcal{B}_1/\mathcal{A}(\mathcal{A}(c))$. Then,

$$\begin{aligned}(\mathcal{B}_2/\mathcal{A})/(\mathcal{B}_1/\mathcal{A})(k) &= (\mathcal{B}_2/\mathcal{A})/(\mathcal{B}_1/\mathcal{A})\circ\mathcal{B}_1/\mathcal{A}(\mathcal{A}(c))\\ &= \mathcal{B}_2/\mathcal{A}\circ\mathcal{A}(c)\\ &= \mathcal{B}_2(c).\end{aligned}$$

Since $\mathcal{B}_1 \leq \mathcal{B}_2$, we have that $\mathcal{B}_2 = \mathcal{B}_2/\mathcal{B}_1 \circ \mathcal{B}_1$. Then,

$$\begin{aligned}\mathcal{B}_2(c) &= \mathcal{B}_2/\mathcal{B}_1 \circ \mathcal{B}_1(c)\\ &= \mathcal{B}_2/\mathcal{B}_1(k),\end{aligned}$$

which concludes the proof. ■

It is often convenient to establish an order on a set of cells. That is, to associate with each cell a distinct integer from 1 to n, where n is the size of that set. We now see that this allows us to represent partitions using matrices.
Consider we identify $\mathcal{C}$ with $\{1,\ldots,|\mathcal{C}|\}$ and $\mathcal{C}/\mathcal{A}$ with $\{1,\ldots,|\mathcal{C}/\mathcal{A}|\}$. Then, we can represent a partition $\mathcal{A}:\mathcal{C}\to\mathcal{C}/\mathcal{A}$ through a **partition matrix** (also called characteristic matrix) $P \in \{0,1\}^{|\mathcal{C}|\times|\mathcal{C}/\mathcal{A}|}$, such that $[P]_{ck} = 1$ if $\mathcal{A}(c) = k$ and $[P]_{ck} = 0$ otherwise. That is, rows corresponds to the cells and columns correspond to the colors, with 1 encoding that the cell of that row maps into the color associated with that column. This is illustrated in the following example.

Example 3.1.7. *Consider the same sets of cells and partitions as in Example 3.1.4. For $\mathcal{C}$ we use the indexing $(a,b,c,d,e) = (1,2,3,4,5)$, for $\mathcal{C}/\mathcal{A}$ we index $(ab,c,de) = (1,2,3)$, and we index $\mathcal{C}/\mathcal{B}$ according to $(ab,cde) = (2,1)$. Note that we indexed ab differently as a member of $\mathcal{C}/\mathcal{A}$ than as a member of $\mathcal{C}/\mathcal{B}$. This is not an issue since an ordering is a property within a given set, not something intrinsic to an element. Using the mentioned indexing, the partitions $\mathcal{A},\mathcal{B},\mathcal{B}/\mathcal{A}$ are represented through the partition matrices $P_\mathcal{A}, P_\mathcal{A}, P_{\mathcal{B}/\mathcal{A}}$, which are given by*

$$P_\mathcal{A} = \begin{bmatrix}1&0&0\\1&0&0\\0&1&0\\0&0&1\\0&0&1\end{bmatrix}, \quad P_\mathcal{B} = \begin{bmatrix}0&1\\0&1\\1&0\\1&0\\1&0\end{bmatrix}, \quad P_{\mathcal{B}/\mathcal{A}} = \begin{bmatrix}0&1\\1&0\\1&0\end{bmatrix}.$$

Using the same indexing, we can also equivalently represent these partitions through the column vectors

$$\mathbf{v}_\mathcal{A} = \begin{bmatrix}1\\1\\2\\3\\3\end{bmatrix}, \quad \mathbf{v}_\mathcal{B} = \begin{bmatrix}2\\2\\1\\1\\1\end{bmatrix}, \quad \mathbf{v}_{\mathcal{B}/\mathcal{A}} = \begin{bmatrix}2\\1\\1\end{bmatrix}.$$

Note that the column vector representation acts as the original partition but with respect to the underlying set orderings. For instance, $\mathcal{A}(d) = de$ is equivalent to $\mathbf{v}_\mathcal{A}(4) = 3$, where we have that d is indexed by 4 in $\mathcal{C}$ and de is indexed by 3 in $\mathcal{C}/\mathcal{A}$. Furthermore, note that the $P_\mathcal{A}$, $\mathbf{v}_\mathcal{A}$ are related such that the c^{th} element of $\mathbf{v}_\mathcal{A}$ indicates the position of the 1 in the c^{th} row of $P_\mathcal{A}$. □

Note that these matrices are related by $P_{\mathcal{A}}P_{\mathcal{B}/\mathcal{A}} = P_{\mathcal{B}}$. This is equivalent to $\mathcal{B}/\mathcal{A} \circ \mathcal{A} = \mathcal{B}$. It is more clear that these formulas are analogous if we consider the transposed version $P^{\top}_{\mathcal{B}/\mathcal{A}}P^{\top}_{\mathcal{A}} = P^{\top}_{\mathcal{B}}$. In this work, we considered it more useful to define partition matrices the way we did instead of the transposed alternative.

Note that given a partition $\mathcal{A}$, we can index its related sets $\mathcal{C}$ and $\mathcal{C}/\mathcal{A}$ in different ways. This means that $\mathcal{A}$ can be represented by multiple partition matrices that are related to each other by a reordering of rows and columns. This is not an issue as long as we keep things consistent by always using the same assigned ordering when constructing other partition matrices that also involve $\mathcal{C}$ and $\mathcal{C}/\mathcal{A}$.

We will often use the partition and its matrix interchangeably, that is, $P_{\mathcal{A}} \leq \mathcal{B}$ or $P_{\mathcal{A}} \leq P_{\mathcal{B}}$ to mean $\mathcal{A} \leq \mathcal{B}$.

Note that given partition matrices $P_{\mathcal{A}}, P_{\mathcal{B}}$ such that $P_{\mathcal{A}} \leq P_{\mathcal{B}}$, we have, from assumption, already assigned an ordering on all the relevant sets $\mathcal{C}$, $\mathcal{C}/\mathcal{A}$ and $\mathcal{C}/\mathcal{B}$. Therefore, there exists an unique partition matrix $P_{\mathcal{AB}}$, representing $\mathcal{B}/\mathcal{A}$ such that $P_{\mathcal{A}}P_{\mathcal{B}/\mathcal{A}} = P_{\mathcal{B}}$.

The trivial partition can be represented by any $|\mathcal{C}| \times |\mathcal{C}|$ permutation matrix, one of which is the identity.

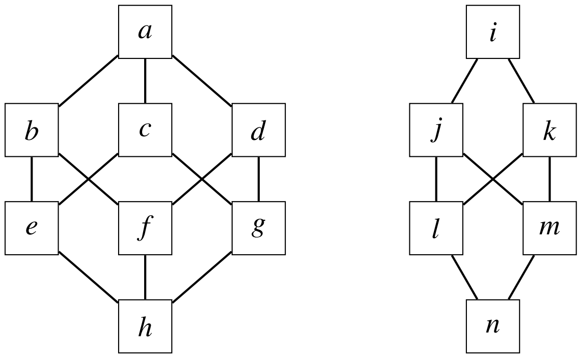
The rank of a partition corresponds to the rank of any of its matrix representations. That is, $rank(\mathcal{A}) = rank(P_{\mathcal{A}})$.

Note that given some matrix M of appropriate dimensions, PM is always well-defined as an expansion of M, where its rows get replicated. In the case of MP, we require the ability of summing elements of M. In our context, the sum operations will be the previously mentioned monoid sum operations $\|$.

3.2 Lattices of partitions

A lattice L is a partially ordered set such that given any two elements $a, b \in L$, there exists in L a **least upper bound** or **join** denoted by $a \vee_L b$. Similarly, there is in L a **greatest lower bound** or **meet** denoted by $a \wedge_L b$.

Example 3.2.1. *Consider Figure 3.1, where we represent two partially ordered sets L and S. We connect two different elements if and only if one is larger than the other (according to its assigned partial order $\leq$) and they have no other element in-between. Furthermore, we present graphically the larger elements above the smaller terms. For instance, in Figure 3.1b, we have that $e \leq_L b$ and $b \leq_L a$ so we connect them. However, we do not connect $e - a$ despite $e \leq_L a$ since b is in-between them. Note that L is a lattice since $\vee_L$ and $\wedge_L$ are well-defined for every pair of elements (e.g., $b \vee_L d = a$ and $b \wedge_L d = f$). On the other hand, S does not have this property. Note that the set of elements larger than l and m is $\{i, j, k\}$. Out of these, j, k are both smaller than i, however, neither $j \leq k$ nor $k \leq j$. That is, they are non-comparable. Since $\{i, j, k\}$ does not have a smallest element, $l \vee_S m$ is not defined, which means that S is not a lattice.* □



(a) Lattice L. (b) Non-lattice S.

Figure 3.1: Partially ordered sets L, S such that L is a lattice and S is not a lattice.

In this work, we are only interested in lattices of partitions, partially ordered according to the finer ($\leq$) relation, described in Equation (3.1).
The set of all partitions on a finite set of cells $\mathcal{C}$, partially ordered by the finer ($\leq$) relation, forms a **lattice** $L_{\mathcal{C}}$. In this set, the join ($\vee$) and meet ($\wedge$) operations can be calculated according to Lemmas 3.2.2 and 3.2.3 respectively.

Lemma 3.2.2. *The partition given by $\mathcal{A} = \mathcal{A}_1 \vee \mathcal{A}_2$ is such that $\mathcal{A}(c) = \mathcal{A}(d)$ if and only if there is a chain of cells $c = c_1, \dots, c_k = d$ such that, for each c_i, c_{i+1}, with $1 \leq i < k$, we have either $\mathcal{A}_1(c_i) = \mathcal{A}_1(c_{i+1})$ or $\mathcal{A}_2(c_i) = \mathcal{A}_2(c_{i+1})$.* □

Proof. Any partition $\mathcal{A}$ that is simultaneous coarser than $\mathcal{A}_1$ and $\mathcal{A}_2$ has to obey (from Equation (3.1))

$$\begin{cases} \mathcal{A}_1(c) & = \mathcal{A}_1(d) \\ & \text{or} \\ \mathcal{A}_2(c) & = \mathcal{A}_2(d) \end{cases} \implies \mathcal{A}(c) = \mathcal{A}(d).$$

For such partition, any chain of cells $c = c_1, \dots, c_k = d$ such that, for each c_i, c_{i+1}, with $1 \leq i < k$, either $\mathcal{A}_1(c_i) = \mathcal{A}_1(c_{i+1})$ or $\mathcal{A}_2(c_i) = \mathcal{A}_2(c_{i+1})$, implies that $\mathcal{A}(c) = \mathcal{A}(d)$. The finest such partition $\mathcal{A}$ is the one such that $\mathcal{A}(c) = \mathcal{A}(d)$ if and only if there is such a chain. Note that the existence of such chains induces an equivalence relation on the set of cells. Therefore, this defines a valid partition. ■

Lemma 3.2.3. *The partition given by $\mathcal{A} = \mathcal{A}_1 \wedge \mathcal{A}_2$ is such that $\mathcal{A}(c) = \mathcal{A}(d)$ if and only if $\mathcal{A}_1(c) = \mathcal{A}_1(d)$ and $\mathcal{A}_2(c) = \mathcal{A}_2(d)$.* □

Proof. Any partition $\mathcal{A}$ that is simultaneous finer than $\mathcal{A}_1$ and $\mathcal{A}_2$ has to obey (from Equation (3.1))

$$\mathcal{A}(c) = \mathcal{A}(d) \implies \begin{cases} \mathcal{A}_1(c) = \mathcal{A}_1(d), \\ \mathcal{A}_2(c) = \mathcal{A}_2(d). \end{cases}$$

The coarsest such partition is created by making the implication into an equivalence. This induces an equivalence relation on the set of cells. Therefore, it defines a valid partition. ■

Not every subset of partitions forms a lattice. Furthermore, subsets of lattices that are themselves lattices might not be sublattices of the original lattice. That is, their join and meet operations might be different. With regard to lattices of partitions, either the join will be coarser that in Lemma 3.2.2 or the meet will be finer than in Lemma 3.2.3 (or both).
Denote by $L_{\mathcal{T}}$ the subset of $L_{\mathcal{C}}$ consisting on the partitions of $\mathcal{C}$ that are finer than $\mathcal{T}$. Note that $L_{\mathcal{T}}$ remains closed under the same join ($\vee$) and meet ($\wedge$) operations. Therefore, $L_{\mathcal{T}}$ is a sublattice of $L_{\mathcal{C}}$.
All the lattices in this work are bounded, which means that they have a (**maximum/greatest element/top**), denoted by $\top$ and a (**minimum/least element/bottom**), denoted by $\bot$. In particular, the top partitions of $L_{\mathcal{C}}$ and $L_{\mathcal{T}}$ are $\top_{\mathcal{C}} = \{\mathcal{C}\}$ and $\top_{\mathcal{T}} = \mathcal{T}$, respectively. The bottom elements $\bot_{\mathcal{C}} = \bot_{\mathcal{T}}$ are given by the trivial partition.
We now show that the existence of a minimal element together with a join operation is enough to guarantee that a finite set forms a lattice.

Lemma 3.2.4. *Consider a finite partially ordered ($\leq_L$) set L such that there is a minimal element $\bot_L \in L$ and for every pair $\mathcal{A}_1, \mathcal{A}_2 \in L$, there exists an element denoted $\mathcal{A}_1 \vee_L \mathcal{A}_2$ which is their least upper bond in L. Then, L is a lattice.* □

Proof. Consider any pair of elements $\mathcal{A}_1, \mathcal{A}_2 \in L$. Call S the subset of L of the elements that are simultaneously smaller ($\leq_L$) than $\mathcal{A}_1$ and $\mathcal{A}_2$. That is, $S := \{\mathcal{P} \in L : \mathcal{P} \leq_L \mathcal{A}_1, \mathcal{A}_2\}$. Note that S is finite. Furthermore, it is not empty since $\bot_L \in S$. Then, to obtain the largest element of S we apply the join ($\vee_L$) operation over the whole set, obtaining $\mathcal{B} = \bigvee^L_{\mathcal{P} \in S} \mathcal{P}$. From assumption, the result is in L. Furthermore, since all the elements of S are smaller than $\mathcal{A}_1$ and $\mathcal{A}_2$, then $\mathcal{B}$ is smaller as well. Therefore, $\mathcal{B} \in S$. By construction, $\mathcal{B}$ is larger than every other element of S, therefore, it is an upper bound of S. That is, $\mathcal{B} \in S$ is the greatest lower bound of $\mathcal{A}_1, \mathcal{A}_2$ in L, which we denote by $\mathcal{A}_1 \wedge_L \mathcal{A}_2$, which means that L is a lattice. ■

In this work, we have particular interest in lattices of partitions L in which the bottom partition is the trivial one ($\bot_L = \bot$) and the join is given according to Lemma 3.2.2 ($\vee_L = \vee$).

Lemma 3.2.5. *Consider a lattice of partitions $L \subseteq L_{\mathcal{T}}$ such that $\bot_L = \bot$ and $\vee_L = \vee$. Then, given any partition $\mathcal{A} \in L_{\mathcal{T}}$, there is a partition $\mathcal{B} \in L$ that is the coarsest one in L such that $\mathcal{B} \leq \mathcal{A}$.* □

Proof. Call S the subset of L of the elements that are finer ($\leq$) than $\mathcal{A}$. That is, $S := \{\mathcal{P} \in L : \mathcal{P} \leq \mathcal{A}\}$. Note that S is finite. Furthermore, it is not empty since $\bot \in S$. Then, to obtain the coarsest element of S we apply the join ($\vee_L = \vee$) operation over the whole set, obtaining $\mathcal{B} = \bigvee_{\mathcal{P} \in S} \mathcal{P}$. Then, $\mathcal{B} \in L$. Furthermore, due to the fact that $L \subseteq L_{\mathcal{T}}$ and $\vee_L = \vee$, we know that all the elements of S being finer than $\mathcal{A}$ implies that $\mathcal{B}$ is finer as well. Therefore, $\mathcal{B} \in S$. By construction, $\mathcal{B}$ is coarser than every other element of S, therefore it is an upper bound of S. That is, $\mathcal{B} \in S$ is the greatest lower bound of $\mathcal{A}$ in L. ■

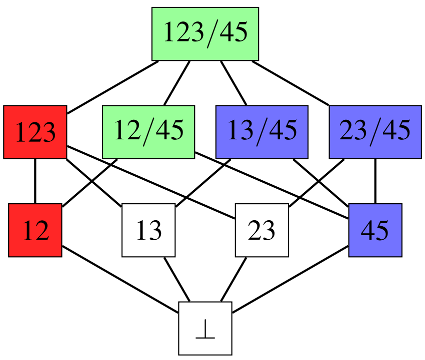
Remark 3.2.6. *Note that Lemma 3.2.5 only holds because we have that* $\perp_L = \perp$ *and* $\vee_L = \vee$*. If* $\perp_L \neq \perp$*, then it would not work for any* $\mathcal{A} < \perp_L$ *(or non-comparable). Furthermore, note that* $\mathcal{A}_1, \mathcal{A}_2 \leq \mathcal{A}$ *only implies* $\mathcal{A}_1 \vee_L \mathcal{A}_2 \leq \mathcal{A}$ *if those partitions are all in the lattice associated with* $\vee_L$*. The fact that* $\vee_L = \vee$ *is what allows us to apply this implication with respect to the lattice* $L_\mathcal{T}$*.* □

The correspondence between partitions $\mathcal{A} \in L_\mathcal{T}$ and $\mathcal{B} \in L$ described in Lemma 3.2.5 establishes a function in $L_\mathcal{T} \to L$, which we denote by cir_L.
We know from Lemma 3.2.4 that a set L with a minimal partition $\perp_L$ and a join $\vee_L$ is automatically a lattice, therefore, it has a meet operation $\wedge_L$. Furthermore, in the case that L is a lattice of partitions such that $\perp_L = \perp$ and $\vee_L = \vee$, it is not guaranteed that $\wedge_L = \wedge$. We know, however, that $\mathcal{A}_1 \wedge_L \mathcal{A}_2 \leq \mathcal{A}_1 \wedge \mathcal{A}_2$. Then, using cir_L, it is clear how to write $\wedge_L$ as a function of $\wedge$.

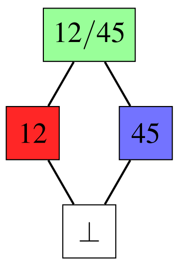
Corollary 3.2.7. *Consider a lattice of partitions* $L \subseteq L_\mathcal{T}$ *such that* $\perp_L = \perp$ *and* $\vee_L = \vee$*. Then, given partitions* $\mathcal{A}_1, \mathcal{A}_2 \in L$*, we have that* $\mathcal{A}_1 \wedge_L \mathcal{A}_2 = cir_L(\mathcal{A}_1 \wedge \mathcal{A}_2)$*.* □

Note that the meet operation $\wedge_L$ is only meaningful when applied to elements of L while cir_L can be applied to any element of $L_\mathcal{T}$.
We now illustrate the cir_L operation in the following example.

Example 3.2.8. *In Figure 3.2a we have the lattice of all partitions finer than* $\mathcal{T} = \{\{1,2,3\},\{4,5\}\}$ *and in Figure 3.2b we have some lattice* L*, which contains the trivial partition* $\perp$ *and is closed under the partition join* $\vee$*. We present the partitions in a simplified manner such that singletons do not appear, which correspond to cells that are not synchronized with any other cell (e.g., the partition* $\{\{1,2\},\{3\},\{4,5\}\}$ *is simply represented as* $12/45$*). The lattices are colored such that each element of* $L_\mathcal{T}$ *is of the same color of the element of* L *that* cir_L *maps to.* □



(a) Lattice $L_\mathcal{T}$.



(b) Lattice L.

Figure 3.2: Illustration of a cir_L function over a suitable lattice of partitions L.

3.3 Lattice quotients

In this section, we define the quotient operation on sets of partitions. In particular, we show that for lattices of partitions L with the properties we are interested in ($\vee_L = \vee$ and $\perp_L = \perp$), all these properties are preserved under the quotient operation.

Definition 3.3.1. *Consider a set of partitions L on some set of cells $\mathcal{C}$. Then, for some $\mathcal{A} \in L$, we define the quotient $L/\mathcal{A}$ as the set of elements of the form $\mathcal{B}/\mathcal{A}$, for all $\mathcal{B} \in L$ such that $\mathcal{A} \leq \mathcal{B}$.* □

Remark 3.3.2. *Note that $L/\mathcal{A}$, which is a set of partitions defined on $\mathcal{C}/\mathcal{A}$, always contains the trivial partition on that set (consider $\mathcal{B} = \mathcal{A}$).* □

We now show that if L is a lattice, then the quotient $L/\mathcal{A}$ is also a lattice in its own right and its join and meet operations are induced from the join and meet of the original lattice L.

Lemma 3.3.3. *Consider a lattice of partitions L and some partition $\mathcal{A} \in L$. Then, $L/\mathcal{A}$ is also a lattice and its join ($\vee_{L/\mathcal{A}}$) and meet ($\wedge_{L/\mathcal{A}}$) operations are given by*

$$(\mathcal{B}_1/\mathcal{A}) \vee_{L/\mathcal{A}} (\mathcal{B}_2/\mathcal{A}) = (\mathcal{B}_1 \vee_L \mathcal{B}_2)/\mathcal{A}, \tag{3.2}$$

$$(\mathcal{B}_1/\mathcal{A}) \wedge_{L/\mathcal{A}} (\mathcal{B}_2/\mathcal{A}) = (\mathcal{B}_1 \wedge_L \mathcal{B}_2)/\mathcal{A}, \tag{3.3}$$

for any $\mathcal{B}_1, \mathcal{B}_2 \in L$ such that $\mathcal{A} \leq \mathcal{B}_1, \mathcal{B}_2$, or equivalently, for any $\mathcal{B}_1/\mathcal{A}, \mathcal{B}_2/\mathcal{A} \in L/\mathcal{A}$. Furthermore, its top ($\top_{L/\mathcal{A}}$) and bottom ($\perp_{L/\mathcal{A}}$) partitions are given by

$$\top_{L/\mathcal{A}} = \top_L/\mathcal{A}, \tag{3.4}$$

$$\perp_{L/\mathcal{A}} = \mathcal{A}/\mathcal{A}. \tag{3.5}$$

□

Proof. Consider any partition $\mathcal{P}/\mathcal{A} \in L/\mathcal{A}$ such that $\mathcal{P}/\mathcal{A} \geq \mathcal{B}_1/\mathcal{A}$ and $\mathcal{P}/\mathcal{A} \geq \mathcal{B}_2/\mathcal{A}$. From Lemma 3.1.5, this is equivalent to saying that $\mathcal{P} \geq \mathcal{B}_1$ and $\mathcal{P} \geq \mathcal{B}_2$. Since $\mathcal{P}, \mathcal{B}_1, \mathcal{B}_2 \in L$, this is equivalent to $\mathcal{P} \geq \mathcal{B}_1 \vee_L \mathcal{B}_2$. Once again from Lemma 3.1.5, this is equivalent to $\mathcal{P}/\mathcal{A} \geq (\mathcal{B}_1 \vee_L \mathcal{B}_2)/\mathcal{A}$. Note that $(\mathcal{B}_1 \vee_L \mathcal{B}_2)/\mathcal{A} \in L/\mathcal{A}$. Furthermore, $(\mathcal{B}_1 \vee_L \mathcal{B}_2)/\mathcal{A}$ is coarser than $\mathcal{B}_1/\mathcal{A}$ and $\mathcal{B}_2/\mathcal{A}$ and any partition that is coarser than them has to also be coarser than $(\mathcal{B}_1 \vee_L \mathcal{B}_2)/\mathcal{A}$. Then, $(\mathcal{B}_1 \vee_L \mathcal{B}_2)/\mathcal{A}$ is the finest such partition, which means that it corresponds to the join ($\vee_{L/\mathcal{A}}$) of $L/\mathcal{A}$, which proves Equation (3.2). Equation (3.3) is proven in a completely analogous way.
Consider any partition $\mathcal{P}/\mathcal{A} \in L/\mathcal{A}$. Then, we have that $\mathcal{P} \in L$ is such that $\mathcal{A} \leq \mathcal{P} \leq \top_L$. Then, from Lemma 3.1.5, we have that $\mathcal{A}/\mathcal{A} \leq \mathcal{P}/\mathcal{A} \leq \top_L/\mathcal{A}$, which proves Equations (3.4) and (3.5). ■

Lemma 3.3.4. *Consider partitions $\mathcal{A}, \mathcal{B}_1, \mathcal{B}_2$ such that $\mathcal{A} \leq \mathcal{B}_1, \mathcal{B}_2$. Then,*

$$(\mathcal{B}_1/\mathcal{A}) \vee (\mathcal{B}_2/\mathcal{A}) = (\mathcal{B}_1 \vee \mathcal{B}_2)/\mathcal{A}. \tag{3.6}$$

□

Proof. Firstly, note that both sides describe partitions on the same set. Assume $\mathcal{A}, \mathcal{B}_1, \mathcal{B}_2$ are partitions on a set of cells $\mathcal{C}$. Then, $\mathcal{B}_1/\mathcal{A}$ and $\mathcal{B}_2/\mathcal{A}$ are partitions on $\mathcal{C}/\mathcal{A}$, and so is their join. Therefore, the left hand side describes a partition on $\mathcal{C}/\mathcal{A}$. It is clear that the right hand side is also a partition on $\mathcal{C}/\mathcal{A}$.
In order to prove that the two partitions are the same, we show that two cells are of the same color in the partition of left hand side if and only if they are also of the same color in the partition of the right hand side. That is,

$$(\mathcal{B}_1/\mathcal{A}) \vee (\mathcal{B}_2/\mathcal{A})(k) = (\mathcal{B}_1/\mathcal{A}) \vee (\mathcal{B}_2/\mathcal{A})(l) \iff$$
$$(\mathcal{B}_1 \vee \mathcal{B}_2)/\mathcal{A}(k) = (\mathcal{B}_1 \vee \mathcal{B}_2)/\mathcal{A}(l)$$

for all $l, k \in \mathcal{C}/\mathcal{A}$. From Lemma 3.2.2, $(\mathcal{B}_1/\mathcal{A}) \vee (\mathcal{B}_2/\mathcal{A})(k) = (\mathcal{B}_1/\mathcal{A}) \vee (\mathcal{B}_2/\mathcal{A})(l)$ is equivalent to the existence of a chain of cells $k = k_1, \ldots, k_n = l$ in $\mathcal{C}/\mathcal{A}$ such that, for each k_i, k_{i+1}, with $1 \leq i < n$, we have either $(\mathcal{B}_1/\mathcal{A})(k_i) = (\mathcal{B}_1/\mathcal{A})(k_{i+1})$ or $(\mathcal{B}_2/\mathcal{A})(k_i) = (\mathcal{B}_2/\mathcal{A})(k_{i+1})$. Note that $k \in \mathcal{C}/\mathcal{A}$ if and only if there is some $c \in \mathcal{C}$ such that $\mathcal{A}(c) = k$. Then, under some cell correspondence $\mathcal{A}(c_i) = k_i$, what we have is equivalent to saying that there is some chain of cells $c = c_1, \ldots, c_n = d$ in $\mathcal{C}$ such that, for each c_i, c_{i+1}, with $1 \leq i < n$, we have either $(\mathcal{B}_1/\mathcal{A})(\mathcal{A}(c_i)) = (\mathcal{B}_1/\mathcal{A})(\mathcal{A}(c_{i+1})$ or $(\mathcal{B}_2/\mathcal{A})(\mathcal{A}(c_i)) = (\mathcal{B}_2/\mathcal{A})(\mathcal{A}(c_{i+1}))$. This simplifies into having that either $\mathcal{B}_1(c_i) = \mathcal{B}_1(c_{i+1})$ or $\mathcal{B}_2(c_i) = \mathcal{B}_2(c_{i+1})$. Then, from Lemma 3.2.2 again, this is equivalent to $\mathcal{B}_1 \vee \mathcal{B}_2(c) = \mathcal{B}_1 \vee \mathcal{B}_2(d)$. Since $\mathcal{A} \leq \mathcal{B}_1, \mathcal{B}_2$ from assumption, it is always true that $\mathcal{A} \leq \mathcal{B}_1 \vee \mathcal{B}_2$. Therefore, what we have is equivalent to $(\mathcal{B}_1 \vee \mathcal{B}_2)/\mathcal{A}(\mathcal{A}(c)) = (\mathcal{B}_1 \vee \mathcal{B}_2)/\mathcal{A}(\mathcal{A}(d))$. This simplifies into $(\mathcal{B}_1 \vee \mathcal{B}_2)/\mathcal{A}(k) = (\mathcal{B}_1 \vee \mathcal{B}_2)/\mathcal{A}(l)$, which completes the proof. ■

The following is now immediate from Lemmas 3.3.3 and 3.3.4.

Corollary 3.3.5. *Consider a lattice of partitions L, some partition $\mathcal{A} \in L$ and its respective quotient lattice $L/\mathcal{A}$. Then, for the joins of those lattices, we have that $\vee_L = \vee$ with regard to partitions coarser than $\mathcal{A}$, if and only if $\vee_{L/\mathcal{A}} = \vee$.* □

In this work we have a particular interest in lattices of partitions that contain the trivial partition and whose join is determined by the partition join of Lemma 3.2.2. We have shown that these properties are preserved under the lattice quotient operation. That is,

Theorem 3.3.6. *Consider a lattice of partitions L on a set of cells $\mathcal{C}$, such that $\bot_L = \bot_{\mathcal{C}}$ and $\vee_L = \vee$. Then, given any partition $\mathcal{A}_1 \in L$, we have that $L/\mathcal{A}$ is a lattice on the set $\mathcal{C}/\mathcal{A}$ such that $\bot_{L/\mathcal{A}} = \bot_{\mathcal{C}/\mathcal{A}}$ and $\vee_{L/\mathcal{A}} = \vee$.* □

We know from Lemma 3.2.5 that lattices with these properties have *cir* functions associated to them. We now show how these functions are related.

Lemma 3.3.7. *Consider a lattice of partitions L such that $\bot_L = \bot$ and $\vee_L = \vee$. Then, given some partition $\mathcal{A} \in L$, the lattice $L/\mathcal{A}$ has a $cir_{L/\mathcal{A}} : L_T/\mathcal{A} \to L/\mathcal{A}$ function, which is related to the*

$cir_L : L_{\mathcal{T}} \to L$ of the original lattice L. In particular, for every $\mathcal{B}/\mathcal{A} \in L_{\mathcal{T}}/\mathcal{A}$, we have that

$$cir_{L/\mathcal{A}}(\mathcal{B}/\mathcal{A}) = cir_L(\mathcal{B})/\mathcal{A}. \tag{3.7}$$

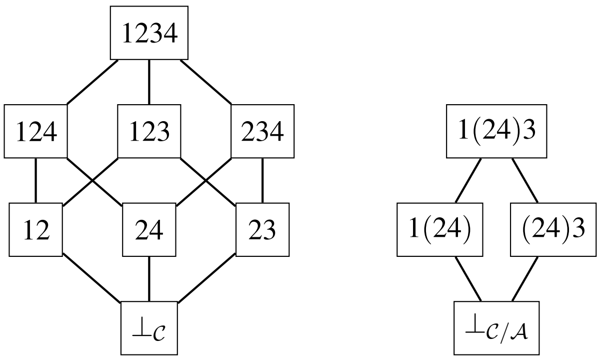
□

Proof. Firstly, note that since we consider elements $\mathcal{B}/\mathcal{A} \in L_{\mathcal{T}}/\mathcal{A}$, we have that $\mathcal{A} \leq \mathcal{B}$ from assumption.

Note that from definition, $cir_L(\mathcal{B})$ is the maximal element of the set $S := \{\mathcal{P} \in L : \mathcal{P} \leq \mathcal{B}\}$ (which we know exists from Lemma 3.2.5). Then, since $\mathcal{A} \in S$, we have that $cir_L(\mathcal{B}) \geq \mathcal{A}$. Therefore, $cir_L(\mathcal{B})/\mathcal{A} \in L/\mathcal{A}$ exists and it corresponds to the maximal element of $S/\mathcal{A}$.

On the other hand, $cir_{L/\mathcal{A}}(\mathcal{B}/\mathcal{A})$ is by definition the maximal term of $\{\mathcal{P}/\mathcal{A} \in L/\mathcal{A} : \mathcal{P}/\mathcal{A} \leq \mathcal{B}/\mathcal{A}\}$, which is again the set $S/\mathcal{A}$, concluding the proof. ■

Example 3.3.8. *In Figure 3.3a we have a lattice of partitions L, on a set of cells $\mathcal{C} = \{1,2,3,4\}$, such that $\bot_L = \bot_{\mathcal{C}}$ and $\vee_L = \vee$. Consider the partition $\mathcal{A} = \{\{1\},\{2,4\},\{3\}\}$, which is in L. We denote the elements of the quotient set $\mathcal{C}/\mathcal{A} = \{1,24,3\}$ and illustrate the quotient lattice $L/\mathcal{A}$ in Figure 3.3b. Note that $L/\mathcal{A}$ is also such that $\bot_{L/\mathcal{A}} = \bot_{\mathcal{C}/\mathcal{A}}$ and $\vee_{L/\mathcal{A}} = \vee$.* □



(a) Lattice L. (b) Quotient lattice $L/\mathcal{A}$.

Figure 3.3: Illustration of a lattice L and its quotient lattice $L/\mathcal{A}$.

3.4 Polydiagonals

We now relate a partition that encodes an equality-based synchrony pattern to its corresponding subset of the state set in the network.

Definition 3.4.1. *Given a partition $\mathcal{A} \in L_{\mathcal{T}}$, we call the subset of $\mathbb{X}$*

$$\Delta_{\mathcal{A}}^{\mathbb{X}} := \{\mathbf{x} \in \mathbb{X} \colon \mathcal{A}(c) = \mathcal{A}(d) \implies x_c = x_d\}, \tag{3.8}$$

*the **polydiagonal** of $\mathcal{A}$ in $\mathbb{X}$.* □

This means that any $\mathbf{x} \in \Delta_{\mathcal{A}}^{\mathbb{X}}$ can be given by $\mathbf{x} = P\bar{\mathbf{x}}$ for some $\bar{\mathbf{x}}$, where P is a partition matrix of $\mathcal{A}$. Consider for instance $\mathcal{A} = \{\{1,2\},\{3\}\}$, represented by $P = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$. Then, $\mathbf{x} = P\bar{\mathbf{x}}$ with $\bar{\mathbf{x}} = \begin{bmatrix} x_{12} \\ x_3 \end{bmatrix}$ gives us $\mathbf{x} = \begin{bmatrix} x_{12} \\ x_{12} \\ x_3 \end{bmatrix}$.

Remark 3.4.2. *Note that if the state sets $\{\mathbb{X}_i\}_{i \in T}$ only have one element, then it is irrelevant to talk about synchronism in the first place. For this reason, we assume that the state sets are non-empty and non-singleton. That is, we can always choose $x_c \neq x_d$ with $x_c, x_d \in \mathbb{X}_i$ for $i = \mathcal{T}(c) = \mathcal{T}(d)$.* □

The partial order relationship between partitions ($\leq$) induces the following inclusion partial order ($\subseteq$) between polydiagonals.

Lemma 3.4.3. *Consider partitions $\mathcal{A}, \mathcal{B} \in L_{\mathcal{T}}$ and their respective polydiagonals $\Delta_{\mathcal{A}}^{\mathbb{X}}, \Delta_{\mathcal{B}}^{\mathbb{X}}$. Then,*

$$\mathcal{A} \leq \mathcal{B} \iff \Delta_{\mathcal{A}}^{\mathbb{X}} \supseteq \Delta_{\mathcal{B}}^{\mathbb{X}}. \tag{3.9}$$

□

Proof. The forward direction is direct from Equation (3.1) together with Definition 3.4.1. The backwards direction is proved by showing its contrapositive, that is, $\neg(\mathcal{A} \leq \mathcal{B}) \implies \neg(\Delta_{\mathcal{A}}^{\mathbb{X}} \supseteq \Delta_{\mathcal{B}}^{\mathbb{X}})$. If $\neg(\mathcal{A} \leq \mathcal{B})$, then there are $c, d \in \mathcal{C}$ such that $\mathcal{A}(c) = \mathcal{A}(d)$ and $\mathcal{B}(c) \neq \mathcal{B}(d)$. Then, under the assumption that the state sets are non-singleton, there is $\mathbf{x} \in \Delta_{\mathcal{B}}^{\mathbb{X}}$ such that $x_c \neq x_d$, that is, $\mathbf{x} \notin \Delta_{\mathcal{A}}^{\mathbb{X}}$, which proves the contrapositive. ■

Moreover, the intersection of two polydiagonals is itself a polydiagonal. In particular, it is related to the join ($\vee$) operation as follows.

Lemma 3.4.4. *Given partitions $\mathcal{A}_1, \mathcal{A}_2 \in L_{\mathcal{T}}$, we have that $\Delta_{\mathcal{A}_1 \vee \mathcal{A}_2}^{\mathbb{X}} = \Delta_{\mathcal{A}_1}^{\mathbb{X}} \cap \Delta_{\mathcal{A}_2}^{\mathbb{X}}$.* □

Proof. Since $\mathcal{A}_1 \vee \mathcal{A}_2$ is coarser than both $\mathcal{A}_1$ and $\mathcal{A}_2$, we know from Lemma 3.4.3 that $\Delta_{\mathcal{A}_1 \vee \mathcal{A}_2}^{\mathbb{X}} \subseteq \Delta_{\mathcal{A}_1}^{\mathbb{X}}$ and $\Delta_{\mathcal{A}_1 \vee \mathcal{A}_2}^{\mathbb{X}} \subseteq \Delta_{\mathcal{A}_2}^{\mathbb{X}}$. Therefore, $\Delta_{\mathcal{A}_1 \vee \mathcal{A}_2}^{\mathbb{X}} \subseteq \Delta_{\mathcal{A}_1}^{\mathbb{X}} \cap \Delta_{\mathcal{A}_2}^{\mathbb{X}}$.
We now prove the converse. Assume $\mathbf{x} \in \Delta_{\mathcal{A}_1}^{\mathbb{X}} \cap \Delta_{\mathcal{A}_2}^{\mathbb{X}}$. Then, $\mathbf{x} \in \Delta_{\mathcal{A}_1}^{\mathbb{X}}$ and $\mathbf{x} \in \Delta_{\mathcal{A}_2}^{\mathbb{X}}$. This implies that for every chain of cells $c = c_1, \ldots, c_k = d$ such that either $\mathcal{A}_1(c_i) = \mathcal{A}_1(c_{i+1})$ or $\mathcal{A}_2(c_i) = \mathcal{A}_2(c_{i+1})$, we have that $x_c = x_d$. From Lemma 3.2.2, we have that $\mathbf{x} \in \Delta_{\mathcal{A}_1 \vee \mathcal{A}_2}^{\mathbb{X}}$. Therefore, $\Delta_{\mathcal{A}_1}^{\mathbb{X}} \cap \Delta_{\mathcal{A}_2}^{\mathbb{X}} \subseteq \Delta_{\mathcal{A}_1 \vee \mathcal{A}_2}^{\mathbb{X}}$. ■

The union of polydiagonals does not necessarily give us another polydiagonal. There exists, however, the smallest polydiagonal that contains the union of two polydiagnals. Note that these properties are analogous to the intersection and union of vector subspaces.

Lemma 3.4.5. *Given partitions $\mathcal{A}_1, \mathcal{A}_2 \in L_{\mathcal{T}}$, we have that $\Delta_{\mathcal{A}}^{\mathbb{X}} \supseteq \Delta_{\mathcal{A}_1}^{\mathbb{X}} \cup \Delta_{\mathcal{A}_2}^{\mathbb{X}}$ if and only if $\mathcal{A} \leq \mathcal{A}_1 \wedge \mathcal{A}_2$.* □

Proof. Consider a partition $\mathcal{A} \in L_{\mathcal{T}}$ such that $\Delta^{\mathbb{X}}_{\mathcal{A}} \supseteq \Delta^{\mathbb{X}}_{\mathcal{A}_1} \cup \Delta^{\mathbb{X}}_{\mathcal{A}_2}$. Then, $\Delta^{\mathbb{X}}_{\mathcal{A}} \supseteq \Delta^{\mathbb{X}}_{\mathcal{A}_1}$ and $\Delta^{\mathbb{X}}_{\mathcal{A}} \supseteq \Delta^{\mathbb{X}}_{\mathcal{A}_2}$. From Lemma 3.4.3, this means that $\mathcal{A} \leq \mathcal{A}_1$ and $\mathcal{A} \leq \mathcal{A}_2$, therefore $\mathcal{A} \leq \mathcal{A}_1 \wedge \mathcal{A}_2$.
We now prove the converse. It is enough to show that $\mathcal{A}_1 \wedge \mathcal{A}_2$ satisfies the inclusion condition since from Lemma 3.4.3, any partition finer than it would also satisfy it. Since $\mathcal{A}_1 \wedge \mathcal{A}_2$ is finer than both $\mathcal{A}_1$ and $\mathcal{A}_2$, we know that $\Delta^{\mathbb{X}}_{\mathcal{A}_1 \wedge \mathcal{A}_2} \supseteq \Delta^{\mathbb{X}}_{\mathcal{A}_1}$ and $\Delta^{\mathbb{X}}_{\mathcal{A}_1 \wedge \mathcal{A}_2} \supseteq \Delta^{\mathbb{X}}_{\mathcal{A}_2}$. Therefore, $\Delta^{\mathbb{X}}_{\mathcal{A}_1 \wedge \mathcal{A}_2} \supseteq \Delta^{\mathbb{X}}_{\mathcal{A}_1} \cup \Delta^{\mathbb{X}}_{\mathcal{A}_2}$. ■

3.5 Invariance of polydiagonals

We now investigate the properties of a function that preserves equality-based synchrony patterns.

Definition 3.5.1. *If for a $\mathcal{G}$-admissible function $f\colon \mathbb{X} \to \mathbb{Y}$ and a partition $\mathcal{A} \in L_{\mathcal{T}}$ we have*

$$f\left(\Delta^{\mathbb{X}}_{\mathcal{A}}\right) \subseteq \Delta^{\mathbb{Y}}_{\mathcal{A}}, \tag{3.10}$$

*then $\mathcal{A}$ is f-**invariant**.*
Furthermore, if for $F \subseteq \mathcal{F}_{\mathcal{G}}$, $\mathcal{A}$ is f-invariant for every $f \in F$, then we say that $\mathcal{A}$ is F-invariant. □

Note that if $\mathcal{A}$ is f-invariant, then for every $\mathbf{x} \in \mathbb{X}$ such that $\mathbf{x} = P\overline{\mathbf{x}}$, with P representing $\mathcal{A}$, there is $\overline{\mathbf{y}}$ such that $f(P\overline{\mathbf{x}}) = P\overline{\mathbf{y}}$. This means that there is a function $\overline{f}\colon \overline{\mathbb{X}} \to \overline{\mathbb{Y}}$ with sets $\overline{\mathbb{X}} := \mathbb{X}^{\overline{\mathbf{k}}}$ and $\overline{\mathbb{Y}} := \mathbb{Y}^{\overline{\mathbf{k}}}$, for an appropriate $\overline{\mathbf{k}} \geq \mathbf{0}_{|T|}$, such that

$$f(P\overline{\mathbf{x}}) = P\overline{f}(\overline{\mathbf{x}}). \tag{3.11}$$

Consider again $\mathcal{A} = \{\{1,2\},\{3\}\}$, represented by $P = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$.

Then, $\mathcal{A}$ is f-invariant if $f\left(\begin{bmatrix} x_{12} \\ x_{12} \\ x_3 \end{bmatrix}\right) = \begin{bmatrix} y_{12} \\ y_{12} \\ y_3 \end{bmatrix}$. That is, $f\left(P\begin{bmatrix} x_{12} \\ x_3 \end{bmatrix}\right) = P\begin{bmatrix} y_{12} \\ y_3 \end{bmatrix}$.

This means that f induces a related function $\overline{f}\left(\begin{bmatrix} x_{12} \\ x_3 \end{bmatrix}\right) = \begin{bmatrix} y_{12} \\ y_3 \end{bmatrix}$.

Consider a discrete-time system $\mathbf{x}^+ = f(\mathbf{x})$ that evolves according to an $\mathcal{G}$-admissible map $f\colon \mathbb{X} \to \mathbb{X}$. If $\mathcal{A}$ is f-invariant, then

$$\mathbf{x}_{n_0} \in \Delta^{\mathbb{X}}_{\mathcal{A}} \implies \mathbf{x}_n \in \Delta^{\mathbb{X}}_{\mathcal{A}} \quad \forall n \in \mathbb{N}\colon n \geq n_0. \tag{3.12}$$

Similarly, for a continuous-time system $\dot{\mathbf{x}} = f(\mathbf{x})$ that evolves according to an $\mathcal{G}$-admissible vector field $f(\mathbf{x})\colon \mathbb{X} \to T_{\mathbf{x}}\mathbb{X}$ where f is Lipschitz, $\mathbb{X}$ is a smooth manifold and $T_{\mathbf{x}}\mathbb{X}$ its tangent space at $\mathbf{x}$. If $\mathcal{A}$ is f-invariant, then

$$\mathbf{x}(t_0) \in \Delta^{\mathbb{X}}_{\mathcal{A}} \implies \mathbf{x}(t) \in \Delta^{\mathbb{X}}_{\mathcal{A}} \quad \forall t \in \mathbb{R}. \tag{3.13}$$

Note that in both cases the polydiagonals $\Delta_{\mathcal{A}}^{\mathbb{X}}$ are invariant with respect to the dynamics. Moreover, the evolution of $\mathbf{x}$ is fully determined by $\overline{\mathbf{x}}$, which in turn evolves according to

$$\overline{\mathbf{x}}^{+}/\dot{\overline{\mathbf{x}}} = \overline{f}(\overline{\mathbf{x}}). \tag{3.14}$$

Corollary 3.5.2. *The trivial partition $\perp$ is always $\mathcal{F}_{\mathcal{G}}$-invariant.* □

Lemma 3.5.3. *Consider partitions $\mathcal{A}_1, \mathcal{A}_2 \in L_{\mathcal{T}}$ and $f \in \mathcal{F}_{\mathcal{G}}$ such that $\mathcal{A}_1, \mathcal{A}_2$ are both f-invariant. Then, $\mathcal{A}_1 \vee \mathcal{A}_2$ is also f-invariant.* □

Proof. Take any $\mathbf{x} \in \Delta_{\mathcal{A}_1 \vee \mathcal{A}_2}^{\mathbb{X}} = \Delta_{\mathcal{A}_1}^{\mathbb{X}} \cap \Delta_{\mathcal{A}_2}^{\mathbb{X}}$. Then, $\mathbf{x} \in \Delta_{\mathcal{A}_1}^{\mathbb{X}}$ and $\mathbf{x} \in \Delta_{\mathcal{A}_2}^{\mathbb{X}}$. From assumption, we have $f(\mathbf{x}) \in \Delta_{\mathcal{A}_1}^{\mathbb{Y}}$ and $f(\mathbf{x}) \in \Delta_{\mathcal{A}_2}^{\mathbb{Y}}$, that is, $f(\mathbf{x}) \in \Delta_{\mathcal{A}_1}^{\mathbb{Y}} \cap \Delta_{\mathcal{A}_2}^{\mathbb{Y}} = \Delta_{\mathcal{A}_1 \vee \mathcal{A}_2}^{\mathbb{Y}}$. Therefore, $\mathcal{A}_1 \vee \mathcal{A}_2$ is f-invariant. ■

Corollary 3.5.4. *Consider partitions $\mathcal{A}_1, \mathcal{A}_2 \in L_{\mathcal{T}}$ and $F \subseteq \mathcal{F}_{\mathcal{G}}$ such that $\mathcal{A}_1, \mathcal{A}_2$ are both F-invariant. Then, $\mathcal{A}_1 \vee \mathcal{A}_2$ is also F-invariant.* □

Proof. From definition, $\mathcal{A}_1, \mathcal{A}_2$ being F-invariant implies that they are f-invariant for every $f \in F$. Then, from Lemma 3.5.3, $\mathcal{A}_1 \vee \mathcal{A}_2$ is also f-invariant for every $f \in F$. That is, $\mathcal{A}_1 \vee \mathcal{A}_2$ is F-invariant. ■

Remark 3.5.5. *Note that only being interested in a particular subset of admissible functions $F \subseteq \mathcal{F}_{\mathcal{G}}$ is quite natural. In particular, the definition of $\mathcal{F}_{\mathcal{G}}$ does not include any type of smoothness assumption. In general, we could be interested in admissible functions that are constructed through of oracle components that have more properties than the minimal ones described in Definition 2.4.1. For instance, an oracle component such that a cell becomes insensitive to cells that are on the same state, corresponds, under the current formalism, to the following constraint.*

$$\hat{f}_i\left(x; \begin{bmatrix} w_{i_1} \\ \mathbf{w} \end{bmatrix}, \begin{bmatrix} x \\ \mathbf{x} \end{bmatrix}\right) = \hat{f}_i\left(x; \mathbf{w}, \mathbf{x}\right). \tag{3.15}$$

This assumption is present, for instance in the Kuramoto model. Note that this makes the cells of such a system always insensitive to self-loops. □

We can now show that the sets of F-invariant partitions form lattices.

Theorem 3.5.6. *Denote by L_F the subset of partitions in $L_{\mathcal{T}}$ that are F-invariant, with $F \subseteq \mathcal{F}_{\mathcal{G}}$. Then, L_F is a lattice whose minimal element $\perp_F$ is the trivial partition $\perp$ and whose join operation $\vee_F$ is the partition join $\vee$ as described in Lemma 3.2.2.* □

Proof. We know that $L_{\mathcal{T}}$ is finite, therefore, L_F is also finite. From Corollary 3.5.2, we know that $\perp \in L_F$ for all $F \subseteq \mathcal{F}_{\mathcal{G}}$. Since $\perp$ is the finest partition, we have that $\perp_F = \perp$.
Consider any $\mathcal{A}_1, \mathcal{A}_2 \in L_F$. Then, from Corollary 3.5.4, we know that $\mathcal{A}_1 \vee \mathcal{A}_2 \in L_F$. Any partition coarser than $\mathcal{A}_1$ and $\mathcal{A}_2$ has to be coarser than $\mathcal{A}_1 \vee \mathcal{A}_2$. Therefore, $\vee_F = \vee$. From Lemma 3.2.4, we know that L_F is a lattice. ■

Remark 3.5.7. *Note that $L_\emptyset = L_\mathcal{T}$ since being $\emptyset$-invariant is vacuously satisfied.* □

Corollary 3.5.8. *Denote by L_f (instead of by $L_{\{f\}}$) the subset of partitions in $L_\mathcal{T}$ that are f-invariant, with $f \in \mathcal{F}_\mathcal{G}$. Then, for all $F \subseteq \mathcal{F}_\mathcal{G}$, we have that $L_F = \bigcap_{f\in F} L_f$.* □

Corollary 3.5.9. *For every $F_1, F_2 \subseteq \mathcal{F}_\mathcal{G}$, we have that*

1. *If $F_1 \subseteq F_2$, then $L_{F_1} \supseteq L_{F_2}$.*
2. *$L_{F_1 \cup F_2} = L_{F_1} \cap L_{F_2}$.*
3. *$L_{F_1 \cap F_2} \supseteq L_{F_1} \cup L_{F_2}$.*

□

From item 1 of Corollary 3.5.9, we know that $L_{\mathcal{F}_\mathcal{G}}$ is the smallest possible lattice of invariant partitions.
We have shown in Lemma 3.2.5 that for a lattice L such that $\perp_L = \perp$ and $\vee_L = \vee$, there exists of a function cir_L that assigns to each element in $L_\mathcal{T}$ an element of L. Since every F-invariant lattice satisfies these assumptions, we have the following.

Corollary 3.5.10. *Consider a F-invariant lattice L_F, with $F \subseteq \mathcal{F}_\mathcal{G}$. Given any partition $\mathcal{A} \in L_\mathcal{T}$, there is a partition $\mathcal{B} \in L_F$ that is the coarsest one in L_F such that $\mathcal{B} \leq \mathcal{A}$. This establishes the function $cir_F : L_\mathcal{T} \to L_F$.* □

Corollary 3.5.11. *Consider partitions $\mathcal{A}_1, \mathcal{A}_2 \in L_F$, with $F \subseteq \mathcal{F}_\mathcal{G}$. Then $\mathcal{A}_1 \wedge_F \mathcal{A}_2 = cir_F(\mathcal{A}_1 \wedge \mathcal{A}_2)$.* □

In summary, we have seen that the join operation $(\vee)$ as described in Lemma 3.2.2 is fundamental with regard to the study of invariance in polydiagonals. In particular, it corresponds to the fact that the intersection of invariant polydiagonals gives us another invariant polydiagonal. On the other hand, the meet operation is not fixed. It is dependent on the particular lattice L and does not present a clear intuitive meaning. In fact, from Lemma 3.2.4, its existence can be seen as a mere consequence of a minimal partition $\perp_L$ together with some join operation $\vee_L$. Since we have that $\vee_L = \vee$ for all the lattices we are interested in (F-invariant lattices), we see that the join operation is the most convenient of the two fundamental operations on lattices and we focus on it in this work.

3.6 Balanced partitions

We now show that if the connectivity structure of a network $\mathcal{G}$ respects certain conditions, it enforces certain polydiagonals to be invariant, regardless of the particular choice of admissible $f \in \mathcal{F}_\mathcal{G}$.

Definition 3.6.1. *Consider a network $\mathcal{G}$ defined on a cell set $\mathcal{C}$ with a cell type partition $\mathcal{T}$ and an in-adjacency matrix M. A partition $\mathcal{A} \in L_{\mathcal{T}}$ with characteristic matrix P is said to be* ***balanced*** *on $\mathcal{G}$ if for all $c,d \in \mathcal{C}$*

$$\mathcal{A}(c) = \mathcal{A}(d) \implies \mathbf{m}_c P = \mathbf{m}_d P, \tag{3.16}$$

where $\mathbf{m}_c, \mathbf{m}_d$ are the rows of matrix M corresponding to cells c and d, respectively. □

Note that a partition is balanced if and only if there is a matrix Q of elements in the appropriate monoids $\{\mathcal{M}_{ij}\}_{i,j \in T}$ such that

$$MP = PQ. \tag{3.17}$$

A balanced partition is usually indicated with the symbol $\bowtie$ and we denote the set of all balanced partitions in a given network $\mathcal{G}$ by $\Lambda_{\mathcal{G}}$.
In Stewart (2007) it was shown that for the unweighted formalism, $\Lambda_{\mathcal{G}}$ forms a lattice under the partition refinement relation ($\leq$), as described in Equation (3.1). We show that this follows easily from the results in Section 3.5.

Corollary 3.6.2. *The trivial partition $\perp$ is always balanced.* □

Proof. For any M, the condition Equation (3.17) is satisfied with $P = I$ and $Q = M$. ■

Lemma 3.6.3. *Consider balanced partitions $\bowtie_1, \bowtie_2 \in \Lambda_{\mathcal{G}}$. Then, $\bowtie_1 \vee \bowtie_2$ is also balanced.* □

Proof. Denote $\bowtie := \bowtie_1 \vee \bowtie_2$ and choose any two colors $A, B \in \bowtie$. Since $\bowtie_1, \bowtie_2$ are both finer than $\bowtie$, there are colors $b_1^1, \ldots, b_{k_1}^1 \in \bowtie_1$ and $b_1^2, \ldots, b_{k_2}^2 \in \bowtie_2$ such that $B = \bigcup_{i=1}^{k_1} b_i^1 = \bigcup_{i=1}^{k_2} b_i^2$. Consider any pair of cells $c, d \in A$. From Lemma 3.2.2, $\bowtie(c) = \bowtie(d)$ implies that there is a chain of cells $c = c_1, \ldots, c_k = d$ such that, for each c_i, c_{i+1}, with $1 \leq i < k$, we have either $\bowtie_1(c_i) = \bowtie_1(c_{i+1})$ or $\bowtie_2(c_i) = \bowtie_2(c_{i+1})$. Then, for each link c_i, c_{i+1} in the chain, there is some $p \in \{1,2\}$ such that

$$\sum_{e \in b_j^p} w_{c_i e} = \sum_{e \in b_j^p} w_{c_{i+1} e} \quad \forall j \in \{1, \ldots, k_p\},$$

which implies

$$\sum_{e \in B} w_{c_i e} = \sum_{e \in B} w_{c_{i+1} e}.$$

Since this quantity is preserved across each link c_i, c_{i+1} of the chain, is it preserved across the whole chain. Therefore,

$$\sum_{e \in B} w_{ce} = \sum_{e \in B} w_{de}$$

for every $c, d \in A$. Since this argument applies to every pair of colors $A, B \in \bowtie$, we have that $\bowtie$ is balanced. ■

Using Lemma 3.2.4 again, the following is an immediate consequence of Corollary 3.6.2 and Lemma 3.6.3.

Corollary 3.6.4. *Given a network $\mathcal{G}$, the set of balanced partitions $\Lambda_{\mathcal{G}}$ forms a lattice whose minimal element $\bot_{\mathcal{G}}$ is the trivial partition $\bot$ and whose join operation $\vee_{\mathcal{G}}$ is the partition join $\vee$ as described in Lemma 3.2.2.* □

From Lemma 3.2.5, the following is immediate.

Corollary 3.6.5. *Given any partition $\mathcal{A} \in L_{\mathcal{T}}$, there is a partition $\bowtie \in \Lambda_{\mathcal{G}}$ that is the coarsest one in $\Lambda_{\mathcal{G}}$ such that $\bowtie \leq \mathcal{A}$.* □

This implies the existence of a *cir* function from $L_{\mathcal{T}}$ to $\Lambda_{\mathcal{G}}$, which we denote by just *cir*. Then, we have the following.

Corollary 3.6.6. *Consider balanced partitions $\bowtie_1, \bowtie_2 \in \Lambda_{\mathcal{G}}$. Then, $\bowtie_1 \wedge_{\mathcal{G}} \bowtie_2 = cir(\bowtie_1 \wedge \bowtie_2)$.* □

The particular *cir* function associated with $\Lambda_{\mathcal{G}}$ is easy to compute and was extended in Sequeira et al. (2021) for the general weighted case.
In order to present the interesting properties of balanced partitions, we require the following result, which relates partitions and oracle components.

Lemma 3.6.7. *For any oracle component $\hat{f}_i \in \hat{\mathcal{F}}_i$, we have that*

$$\hat{f}_i(x; \mathbf{w}, P\overline{\mathbf{x}}) = \hat{f}_i(x; P^\top \mathbf{w}, \overline{\mathbf{x}}), \tag{3.18}$$

where P is a partition matrix of appropriate dimensions such that the vectors $\mathbf{w}$ and $P\overline{\mathbf{x}}$ have elements of matching cell types. □

Proof. We prove this by induction. Consider fixed integers n, k such that $0 < k < n$. Assume Equation (3.18) applies to all partition matrices of dimension $n \times (k+1)$ as long as it is applied to suitable (type matching) $\mathbf{w}$ and $\overline{\mathbf{x}}$. Note that any partition matrix P of dimension $n \times k$ can be obtained by taking some partition matrix $\overline{P}$ of dimension $n \times (k+1)$ and merging together two of its columns. That is, $P = \overline{P} p \sigma$, with $p = \begin{bmatrix} 1 & \mathbf{0}^\top \\ 1 & \mathbf{0}^\top \\ \mathbf{0} & \mathbf{I}_{k-1} \end{bmatrix}$ and where σ is a permutation matrix of dimension $k \times k$.
Consider one such P and any suitable $\mathbf{w}$ and $\overline{\mathbf{x}}$. Then, $\hat{f}_i(x; \mathbf{w}, P\overline{\mathbf{x}}) = \hat{f}_i(x; \mathbf{w}, \overline{P}(p\sigma\overline{\mathbf{x}}))$. From assumption, we call apply Equation (3.18) with respect to $\overline{P}$, which gets us $\hat{f}_i(x; \overline{P}^\top \mathbf{w}, p\sigma\overline{\mathbf{x}})$. Due to the particular shape of p, applying Equation (3.18) with regard to p is equivalent to item 2 of Definition 2.4.1. This gives us $\hat{f}_i(x; p^\top \overline{P}^\top \mathbf{w}, \sigma\overline{\mathbf{x}})$. Similarly, we can apply Equation (3.18) with regard to σ since it corresponds to item 1 of Definition 2.4.1. Note that since σ is a permutation matrix, we have that $\sigma^{-1} = \sigma^\top$. Therefore, this becomes $\hat{f}_i(x; \sigma^\top p^\top \overline{P}^\top \mathbf{w}, \overline{\mathbf{x}}) = \hat{f}_i(x; (\overline{P} p \sigma)^\top \mathbf{w}, \overline{\mathbf{x}}) = \hat{f}_i(x; P^\top \mathbf{w}, \overline{\mathbf{x}})$, which proves that Equation (3.18) is satisfied for any partition matrix P of size $n \times k$.

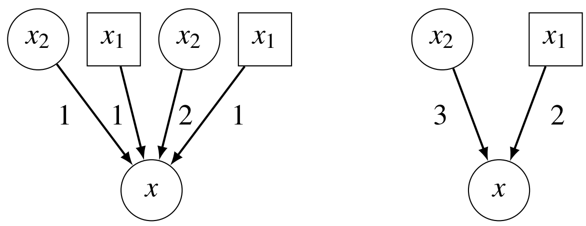
In the base case $k = n$, the partition matrix is in fact a permutation, therefore, it is direct from item 1 of Definition 2.4.1, which concludes the proof. ■

Remark 3.6.8. *Note that Lemma 3.6.7 is valid for all inputs such that the evaluation is meaningful. That is, whenever the domain Equation (2.1) is respected. Furthermore, it can be seen that vectors* $\mathbf{w}$ *and* $P\overline{\mathbf{x}}$ *having elements of matching cell types is equivalent to* $P^\top \mathbf{w}$ *and* $\overline{\mathbf{x}}$ *having elements of matching cell types and the sum* $P^\top \mathbf{w}$ *being well-defined. That is, each sum operates on elements of the same commutative monoid.* □

Remark 3.6.9. *We just proved Lemma 3.6.7 using items 1 and 2 in Definition 2.4.1. Furthermore, it is straightforward that these items are just particular cases of Lemma 3.6.7. Therefore, these are equivalent statements.* □

This result is illustrated in the following example.

Example 3.6.10. *Consider the networks in Figure 3.4, consisting on a cell and its respective in-neighborhood. We have cell types* $T = \{1,2\}$ *which represent "circle" and "square" cells, respectively. We define the monoid operations* $\|$ *to be the usual addition. If we write the state and*



(a) First input set. (b) Second (edge compressed) input set.

Figure 3.4: Networks with equivalent input sets.

weight vectors from left to right, we get

$$\mathbf{x} = \begin{bmatrix} x_2 & x_1 & x_2 & x_1 \end{bmatrix}^\top, \qquad \mathbf{w} = \begin{bmatrix} 1 & 1 & 2 & 1 \end{bmatrix}^\top,$$
$$\overline{\mathbf{x}} = \begin{bmatrix} x_2 & x_1 \end{bmatrix}^\top, \qquad \overline{\mathbf{w}} = \begin{bmatrix} 3 & 2 \end{bmatrix}^\top$$

for each of the Figures 3.4a and 3.4b, respectively. Consider now the partition matrix

$$P = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Note that $\mathbf{x} = P\overline{\mathbf{x}}$ *and* $\overline{\mathbf{w}} = P^{\top}\mathbf{w}$*. Furthermore, these operations respect the cell types since we have that* $\mathcal{T} = P\overline{\mathcal{T}}$*, with*

$$\mathcal{T} = \begin{bmatrix} 1 & 2 & 1 & 2 \end{bmatrix}^{\top}, \qquad \overline{\mathcal{T}} = \begin{bmatrix} 1 & 2 \end{bmatrix}^{\top}.$$

Then, from Lemma 3.6.7, we see that

$$\hat{f}_i(x;\mathbf{w},\mathbf{x}) = \hat{f}_i(x;\mathbf{w},P\overline{\mathbf{x}}) = \hat{f}_i(x;P^{\top}\mathbf{w},\overline{\mathbf{x}}) = \hat{f}_i(x;\overline{\mathbf{w}},\overline{\mathbf{x}}).$$

□

Having proven this, we can now state the following result, which underlines the importance of balanced partitions in the study of invariance.

Theorem 3.6.11. *Consider a balanced partition* $\bowtie \in \Lambda_{\mathcal{G}}$ *on a network* $\mathcal{G}$ *and any* $\mathcal{G}$*-admissible function* $f \in \mathcal{F}_{\mathcal{G}}$*. Then,* $\bowtie$ *is* f*-invariant.* □

Proof. Consider any $\bowtie \in \Lambda_{\mathcal{G}}$ and a state in the related polydiagonal $\mathbf{x} \in \Delta_{\bowtie}^{\mathbb{X}}$. That is, $\mathbf{x} = P\overline{\mathbf{x}}$ for some $\overline{\mathbf{x}}$, where P is a partition matrix of $\bowtie$.
For any pair of cells $c,d \in \mathcal{C}$ such that $\bowtie(c) = \bowtie(d)$, we have that $x_c = x_d$. Furthermore, from Definition 3.6.1, we have that $P^{\top}\mathbf{m}_c^{\top} = P^{\top}\mathbf{m}_d^{\top}$. Therefore, $\hat{f}_i\left(x_c;P^{\top}\mathbf{m}_c^{\top},\overline{\mathbf{x}}\right) = \hat{f}_i\left(x_d;P^{\top}\mathbf{m}_d^{\top},\overline{\mathbf{x}}\right)$ for any $\hat{f}_i \in \hat{\mathcal{F}}_i$, with $i = \mathcal{T}(c) = \mathcal{T}(d)$.
Using Lemma 3.6.7, this becomes $\hat{f}_i\left(x_c;\mathbf{m}_c^{\top},P\overline{\mathbf{x}}\right) = \hat{f}_i\left(x_d;\mathbf{m}_d^{\top},P\overline{\mathbf{x}}\right)$, which from Definition 2.4.5 is equivalent to $f_c(P\overline{\mathbf{x}}) = f_d(P\overline{\mathbf{x}})$. This means that for every $\mathcal{G}$-admissible function $f \in \mathcal{F}_{\mathcal{G}}$, there is a $\overline{f}$ such that $f(P\overline{\mathbf{x}}) = P\overline{f}(\overline{\mathbf{x}})$. That is, $\bowtie$ is f-invariant. ■

Which is equivalent to the following statement.

Corollary 3.6.12. *Given a network* $\mathcal{G}$*, we have that* $\Lambda_{\mathcal{G}} \subseteq L_F$*, for any* $F \subseteq \mathcal{F}_{\mathcal{G}}$*.* □

Theorem 3.6.13. *Consider a partition* $\mathcal{A} \leq \mathcal{T}$ *on some network* $\mathcal{G}$*. If* $\mathcal{A}$ *is* $\mathcal{F}_{\mathcal{G}}$*-invariant, then* $\mathcal{A}$ *is balanced on* $\mathcal{G}$*.* □

Proof. We prove this through its contrapositive. That is, $\mathcal{A} \notin \Lambda_{\mathcal{G}}$ implies $\mathcal{A} \notin L_{\mathcal{F}_{\mathcal{G}}}$.
Consider any partition $\mathcal{A} \leq \mathcal{T}$ that is not balanced. Then, there are cells $c,d \in \mathcal{C}$ such that $\mathcal{A}(c) = \mathcal{A}(d)$ and $P^{\top}\mathbf{m}_c^{\top} \neq P^{\top}\mathbf{m}_d^{\top}$, where P is a partition matrix of $\mathcal{A}$. Consider k one of the entries in which they differ. That is, $\left[P^{\top}\mathbf{m}_c^{\top}\right]_k \neq \left[P^{\top}\mathbf{m}_d^{\top}\right]_k$.
We now choose a state in the related polydiagonal $\mathbf{x} \in \Delta_{\mathcal{A}}^{\mathbb{X}}$, that is, $\mathbf{x} = P\overline{\mathbf{x}}$ for some $\overline{\mathbf{x}}$, such that $\overline{x}_k$ is different from all other entries of $\overline{\mathbf{x}}$.
Then, there is an $f \in \mathcal{F}_{\mathcal{G}}$, that is, $f = \hat{f}|_{\mathcal{G}}$ such that $\hat{f}_i(x;\mathbf{w},\mathbf{x}) = y_1$ if $\mathbf{w}$, summed over the entries in $\mathbf{x}$ that are $\overline{x}_k$, results into $\left[P^{\top}\mathbf{m}_c^{\top}\right]_k$, and $\hat{f}_i(x;\mathbf{w},\mathbf{x}) = y_2$ otherwise, with $y_1 \neq y_2$, $y_1,y_2 \in \mathbb{Y}_i$ and $i = \mathcal{T}(c)$. Then, $\hat{f}_i\left(x_c;P^{\top}\mathbf{m}_c^{\top},\overline{\mathbf{x}}\right) = y_1$ and $\hat{f}_i\left(x_d;P^{\top}\mathbf{m}_d^{\top},\overline{\mathbf{x}}\right) = y_2$. That is, $f_c(P\overline{\mathbf{x}}) \neq f_d(P\overline{\mathbf{x}})$, and we have an $f \in \mathcal{F}_{\mathcal{G}}$ and $\overline{\mathbf{x}}$ such that $f(P\overline{\mathbf{x}}) \notin \Delta_{\mathcal{A}}^{\mathbb{Y}}$. That is, $\mathcal{A} \notin \Lambda_{\mathcal{G}}$ implies $\mathcal{A} \notin L_{\mathcal{F}_{\mathcal{G}}}$, which completes the proof. ■

Which is equivalent to the following statement.

Corollary 3.6.14. *Given a network $\mathcal{G}$, we have that $\Lambda_{\mathcal{G}} \supseteq L_{\mathcal{F}_{\mathcal{G}}}$.* □

Note that $L_{\mathcal{F}_{\mathcal{G}}}$ is the smallest possible lattice of invariant partitions. In Stewart et al. (2003); Golubitsky et al. (2005); Golubitsky and Stewart (2006), Theorem 3.6.13 was derived by proving a stronger result. In particular, by showing that exists some subset $F \subseteq \mathcal{F}_{\mathcal{G}}$ such that $\Lambda_{\mathcal{G}} \supseteq L_F$. This type of results is of interest since one might only be interested in certain subclasses of admissible functions and not the full $\mathcal{F}_{\mathcal{G}}$.
This stronger result, which was originally hid away in their proof of the unweighted version of Theorem 3.6.13, was made explicit and generalized in Sequeira et al. (2021) for the general weighted formalism.
From and Corollaries 3.6.12 and 3.6.14 the following is now immediate.

Corollary 3.6.15. *Given a network $\mathcal{G}$, we have that $\Lambda_{\mathcal{G}} = L_{\mathcal{F}_{\mathcal{G}}}$.* □

3.7 Quotient networks

In this section we describe how the behavior of a network $\mathcal{G}$ when evaluated at some polydiagonal $\Delta_{\bowtie}^{\mathbb{X}}$ for some balanced partition $\bowtie$ can be described by a smaller network $\mathcal{Q}$.

Definition 3.7.1. *Consider a network $\mathcal{G}$ defined on a cell set $\mathcal{C}_{\mathcal{G}}$ with a cell type partition $\mathcal{T}_{\mathcal{G}}$ and an in-adjacency matrix M. Take a balanced partition $\bowtie \in \Lambda_{\mathcal{G}}$.*
*The **quotient network** $\mathcal{Q}$ of $\mathcal{G}$ over $\bowtie$, denoted $\mathcal{Q} := \mathcal{G}/\bowtie$, is defined on a cell set $\mathcal{C}_{\mathcal{Q}} := \mathcal{C}_{\mathcal{G}}/\bowtie$ with a cell type partition $\mathcal{T}_{\mathcal{Q}} := \mathcal{T}_{\mathcal{G}}/\bowtie$ and an in-adjacency matrix Q given by $MP = PQ$, where P represents $\bowtie$.* □

Remark 3.7.2. *We assume that a particular ordering has been chosen for the sets of cells $\mathcal{C}_{\mathcal{G}}$ and $\mathcal{C}_{\mathcal{Q}}$. Then, the partition P representing $\bowtie$ and the in-adjacency matrices M and Q are uniquely defined.* □

Lemma 3.7.3. *Consider a balanced partition $\bowtie_{01} \in \Lambda_{\mathcal{G}_0}$ on a network $\mathcal{G}_0$ and its respective quotient network $\mathcal{G}_1 = \mathcal{G}_0/\bowtie_{01}$. For a partition $\bowtie_{02}$ such that $\bowtie_{01} \leq \bowtie_{02}$, define $\bowtie_{12} := \bowtie_{02}/\bowtie_{01}$. Then, $\bowtie_{02} \in \Lambda_{\mathcal{G}_0}$ if and only if $\bowtie_{12} \in \Lambda_{\mathcal{G}_1}$. Furthermore, if $\bowtie_{02}$ and $\bowtie_{12}$ satisfy this, then $\mathcal{G}_0/\bowtie_{02} = \mathcal{G}_1/\bowtie_{12}$.* □

Proof. Denote by $\mathcal{T}_0$, $\mathcal{T}_1$, the cell type partitions of networks $\mathcal{G}_0$, $\mathcal{G}_1$, respectively. Then, $\mathcal{T}_1 = \mathcal{T}_0/\bowtie_{01}$ by definition. From Lemma 3.1.6, we know that $\mathcal{T}_1/\bowtie_{12} = (\mathcal{T}_0/\bowtie_{01})/(\bowtie_{02}/\bowtie_{01}) = \mathcal{T}_0/\bowtie_{02}$.
Consider now that M_0, M_1 are the in-adjacency matrices of $\mathcal{G}_0$, $\mathcal{G}_1$, respectively, and that P_{01}, P_{02}, P_{12} are the partition matrices of $\bowtie_{01}$, $\bowtie_{02}$, $\bowtie_{12}$. Then, from $\bowtie_{12} = \bowtie_{02}/\bowtie_{01}$, we have that $P_{02} = P_{01}P_{12}$. Moreover, since $\bowtie_{01} \in \Lambda_{\mathcal{G}_0}$, we have that $M_0P_{01} = P_{01}M_1$.
In order to show that $\bowtie_{02} \in \Lambda_{\mathcal{G}_0}$ if and only if $\bowtie_{12} \in \Lambda_{\mathcal{G}_1}$, we prove that

$$M_0P_{02} = P_{02}M_2 \iff M_1P_{12} = P_{12}M_2.$$

Expanding P_{02} in the left hand side, this becomes

$$M_0P_{01}P_{12} = P_{01}P_{12}M_2 \iff M_1P_{12} = P_{12}M_2.$$

From $M_0P_{01} = P_{01}M_1$, this can be written as

$$P_{01}(M_1P_{12}) = P_{01}(P_{12}M_2) \iff M_1P_{12} = P_{12}M_2,$$

which is now clear from the fact that P_{01} has full column rank, that is, it is left invertible. Note that if there is a matrix M_2 that satisfies these expressions, the network $\mathcal{G}_2$ defined by the in-adjacency matrix M_2 and the cell type partition $\mathcal{T}_2 = \mathcal{T}_1/\bowtie_{12} = \mathcal{T}_0/\bowtie_{02}$, is such that $\mathcal{G}_2 = \mathcal{G}_0/\bowtie_{02} = \mathcal{G}_1/\bowtie_{12}$. ■

From Theorem 3.6.11, we know that any balanced partition $\bowtie \in \Lambda_{\mathcal{G}}$ is f-invariant for any $f \in \mathcal{F}_{\mathcal{G}}$. Note that in Equation (3.11) it was shown that for a partition $\mathcal{A}$ and a function f such that $\mathcal{A}$ is f-invariant, then, f, when evaluated on $\Delta^{\mathbb{X}}_{\mathcal{A}}$ can be determined by a simpler function $\overline{f}$. We will see that for the case of balanced partitions this function is particularly noteworthy.

Definition 3.7.4. *Consider a network $\mathcal{G}$ and a balanced partition $\bowtie \in \Lambda_{\mathcal{G}}$. Let $f \in \mathcal{F}_{\mathcal{G}}$. The **quotient function** $g := f/\bowtie$ is defined through constraining f to the polydiagonal $\Delta^{\mathbb{X}}_{\bowtie}$. That is,*

$$f(P\overline{\mathbf{x}}) = Pg(\overline{\mathbf{x}}), \tag{3.19}$$

where P is the partition matrix of $\bowtie$. □

We now show that the quotient function is very intimately related to the quotient network.

Theorem 3.7.5. *Consider networks $\mathcal{G}$ and $\mathcal{Q}$ such that $\mathcal{Q} = \mathcal{G}/\bowtie$ for some balanced partition $\bowtie \in \Lambda_{\mathcal{G}}$. Then, for any $f \in \mathcal{F}_{\mathcal{G}}$, which is given by $f = \hat{f}|_{\mathcal{G}}$, for some $\hat{f} \in \hat{\mathcal{F}}_T$, we have that its quotient function $g = f/\bowtie$ is given by $g = \hat{f}|_{\mathcal{Q}}$. Therefore, $g \in \mathcal{F}_{\mathcal{Q}}$.* □

Proof. From Definition 3.7.4, we have that $g_k(\overline{\mathbf{x}}) = f_c(P\overline{\mathbf{x}})$ for all $c \in \mathcal{C}_{\mathcal{G}}$ and $k = \bowtie(c)$. Then, from $\mathcal{G}$-admissibility, $f_c(P\overline{\mathbf{x}}) = \hat{f}_i\left(\overline{x}_k; \mathbf{m}_c^\top, P\overline{\mathbf{x}}\right)$, with $i = \mathcal{T}_{\mathcal{G}}(c)$. From Lemma 3.6.7, $\hat{f}_i\left(\overline{x}_k; \mathbf{m}_c^\top, P\overline{\mathbf{x}}\right) = \hat{f}_i\left(\overline{x}_k; P^\top\mathbf{m}_c^\top, \overline{\mathbf{x}}\right)$, and since $\bowtie$ is balanced, this is equal to $\hat{f}_i(\overline{x}_k; \mathbf{q}_k^\top, \overline{\mathbf{x}})$. That is, we have that $g_k(\overline{\mathbf{x}}) = \hat{f}_i(\overline{x}_k; \mathbf{q}_k^\top, \overline{\mathbf{x}})$ for all $k \in \mathcal{C}_{\mathcal{Q}}$ with $i = \mathcal{T}_{\mathcal{Q}}(k)$. Therefore, $g = (g_k)_{k \in \mathcal{C}_{\mathcal{Q}}}$ is $\mathcal{Q}$-admissible, with $g = \hat{f}|_{\mathcal{Q}}$. ■

The following is now immediate from Definition 3.7.4 and Theorem 3.7.5.

Corollary 3.7.6. *Consider networks $\mathcal{G}$ and $\mathcal{Q}$ such that $\mathcal{Q} = \mathcal{G}/\bowtie$ for some balanced partition $\bowtie \in \Lambda_{\mathcal{G}}$. Then, for any $\hat{f}^1, \hat{f}^2 \in \hat{\mathcal{F}}_T$, we have that*

$$\hat{f}^1|_{\mathcal{G}} = \hat{f}^2|_{\mathcal{G}} \implies \hat{f}^1|_{\mathcal{Q}} = \hat{f}^2|_{\mathcal{Q}}.$$

□

Now that we understand the relationship between $f \in \mathcal{F}_\mathcal{G}$ and its quotient $g = f/\bowtie$ in terms of oracle functions, the following is clear.

Corollary 3.7.7. *Consider networks $\mathcal{G}$ and $\mathcal{Q}$ such that $\mathcal{Q} = \mathcal{G}/\bowtie$ for some balanced partition $\bowtie \in \Lambda_\mathcal{G}$. Then, for any $g \in \mathcal{F}_\mathcal{Q}$, there is some $f \in \mathcal{F}_\mathcal{G}$ such that $g = f/\bowtie$.* □

Note that Corollary 3.7.7 only refers to existence, not to uniqueness. That is, it could be possible to have $f_1, f_2 \in \mathcal{F}_\mathcal{G}$ such that $f_1 \neq f_2$ but $g = f_1/\bowtie = f_2/\bowtie$. They will, however, match when evaluated at the polydiagonal $\Delta_\bowtie^\mathbb{X}$.

Example 3.7.8. *Consider the given partition $\mathcal{A} = \{\{1,2\},\{3\}\}$ on the* **CCN** *of Example 2.4.6 (Figure 2.6). One partition matrix of $\mathcal{A}$ is*

$$P = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{3.20}$$

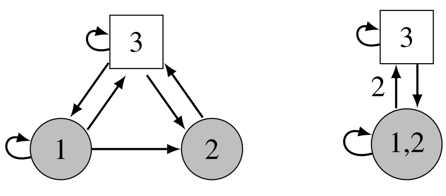
in which each column identifies one of the colors of the partition. From this we obtain the product

$$MP = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 2 & 1 \end{bmatrix}. \tag{3.21}$$

Note that rows 1 and 2 are the same and the respective cells are of the same cell type. That means that for any admissible f we have $f_1(\mathbf{x}) = f_2(\mathbf{x})$ when $x_1 = x_2$.
Observe that this is in agreement with the functional form we wrote in Equations (2.7) and (2.8).
Since the rows of MP respect an equality relationship according to $\mathcal{A}$, then $\mathcal{A}$ is balanced and there is a quotient matrix Q that obeys the balanced condition Equation (3.17). In fact, the quotient matrix Q is

$$Q = \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix}, \tag{3.22}$$

which is directly obtained from MP by compressing its rows according to $\mathcal{A}$.
The behavior of this **CCN** *when $x_1 = x_2$ is then described by the smaller* **CCN** *given by the quotient matrix Q which is represented in Figure 3.5b. The coloring is a way of representing the partition*



(a) Original. (b) Quotient.

Figure 3.5: Color-coded network of Figure 2.6 and its quotient over the balanced partition $\{\{1,2\},\{3\}\}$.

$\mathcal{A} = \{\{1,2\},\{3\}\}$ over which the quotient is done. Note that in both Figures 3.5a and 3.5b each

gray cell receives one connection from a gray cell and one connection from a white cell. On the other hand, each white cell receives a connection from a white cell and two connections from a gray cell. The function $g = f/\bowtie$ *has the following structure*

$$g_{12}(\mathbf{x}) = \hat{f}_1(x_{12}; \begin{bmatrix} 1 & 1 \end{bmatrix}^\top, \mathbf{x}), \tag{3.23}$$

$$g_3(\mathbf{x}) = \hat{f}_2(x_3; \begin{bmatrix} 2 & 1 \end{bmatrix}^\top, \mathbf{x}), \tag{3.24}$$

where $\hat{f} \in \hat{\mathcal{F}}_T$ *is any oracle function such that* $f = \hat{f}|_{\mathcal{G}}$*.* □

Remark 3.7.9. *Note that finding a balanced partition from its graph representation or its matrix* M *is not obvious. See Example 3.7.10.* □

Example 3.7.10. *Consider the following network illustrated in Figure 3.6. Since cells* 2 *and* 3 *have*

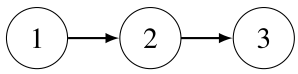


Figure 3.6: Chain **CCN**.

the same type of input it might be tempting to think that $\mathcal{A} = \{\{1\}, \{2, 3\}\}$ *should be balanced. Note, however, that the rows of the corresponding matrix* MP *Equation (3.25) do not respect the row equalities according to* $\mathcal{A}$*, which means that it is not balanced.*

$$MP = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{3.25}$$

Another way to see this is to color the cells according to the partition (Figure 3.7) and see that cells with the same color do not have equivalent colored input sets. Note that cells 2 *and* 3 *are*

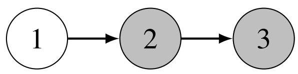


Figure 3.7: Unbalanced coloring.

both gray but one of them receives one edge from a white cell and the other receives one edge from a gray cell. Therefore, this coloring (partition) is not balanced. In fact, it can be easily seen that the only balanced partition of this network is the trivial one. □

We now extend the concept of quotient of admissible functions to sets of admissible functions.

Definition 3.7.11. *Consider networks* $\mathcal{G}$ *and* $\mathcal{Q}$ *such that* $\mathcal{Q} = \mathcal{G}/\bowtie$ *for some* $\bowtie \in \Lambda_{\mathcal{G}}$*. Given any subset of* $\mathcal{G}$*-admissible functions* $F_{\mathcal{G}} \subseteq \mathcal{F}_{\mathcal{G}}$*, we define its quotient* $F_{\mathcal{Q}} = F_{\mathcal{G}}/\bowtie$ *as the subset of* $\mathcal{F}_{\mathcal{Q}}$ *such that* $g \in F_{\mathcal{Q}}$ *if and only if there is some* $f \in F_{\mathcal{G}}$ *such that* $g = f/\bowtie$*.* □

Note that from Corollary 3.7.7 it is immediate that $\mathcal{F}_\mathcal{G}/\bowtie = \mathcal{F}_\mathcal{Q}$. That is, $\mathcal{F}_\mathcal{G}/\bowtie = \mathcal{F}_{\mathcal{G}/\bowtie}$.
We are now ready to study the relation between the invariant lattices $L_{F_\mathcal{G}}$ of a network $\mathcal{G}$ and corresponding invariant lattice of its quotient network $\mathcal{Q} = \mathcal{G}/\bowtie$.
The following is direct from Lemma 3.7.3.

Corollary 3.7.12. *Consider networks $\mathcal{G}$ and $\mathcal{Q}$ such that $\mathcal{Q} = \mathcal{G}/\bowtie$ for some $\bowtie \in \Lambda_\mathcal{G}$. Then, $\Lambda_\mathcal{Q} = \Lambda_\mathcal{G}/\bowtie$.* □

We now generalize this to lattices of F-invariant partitions.

Theorem 3.7.13. *Consider networks $\mathcal{G}$ and $\mathcal{Q}$ such that $\mathcal{Q} = \mathcal{G}/\bowtie$ for some $\bowtie \in \Lambda_\mathcal{G}$, and subsets $F_\mathcal{G} \subseteq \mathcal{F}_\mathcal{G}$, $F_\mathcal{Q} \subseteq \mathcal{F}_\mathcal{Q}$ such that $F_\mathcal{Q} = F_\mathcal{G}/\bowtie$.*
Then, for partitions $\mathcal{A}_\mathcal{G} \leq \mathcal{T}_\mathcal{G}$ and $\mathcal{A}_\mathcal{Q} \leq \mathcal{T}_\mathcal{Q}$ such that $\mathcal{A}_\mathcal{G} \geq \bowtie$ and $\mathcal{A}_\mathcal{Q} = \mathcal{A}_\mathcal{G}/\bowtie$, we have that $\mathcal{A}_\mathcal{G} \in L_{F_\mathcal{G}}$ if and only if $\mathcal{A}_\mathcal{Q} \in L_{F_\mathcal{Q}}$. That is, $L_{F_\mathcal{Q}} = L_{F_\mathcal{G}}/\bowtie$. □

Proof. We consider the partitions $\bowtie$, $\mathcal{A}_\mathcal{G}$ and $\mathcal{A}_\mathcal{Q}$ to be represented by partition matrices P, $P_\mathcal{G}$ and $P_\mathcal{Q}$ respectively, such that $P_\mathcal{G} = PP_\mathcal{Q}$.
Assume $\mathcal{A}_\mathcal{G} \in L_{F_\mathcal{G}}$. Note that for any $g \in F_\mathcal{Q}$, there is some $f \in F_\mathcal{G}$ such that $g = f/\bowtie$. Then, from the fact that $\bowtie$ is balanced, we know from Definition 3.7.4 that $f(P_\mathcal{G}\overline{\mathbf{x}}) = f(PP_\mathcal{Q}\overline{\mathbf{x}}) = Pg(P_\mathcal{Q}\overline{\mathbf{x}})$. On the other hand, from the fact that $\mathcal{A}_\mathcal{G} \in L_{F_\mathcal{G}}$ we know that $f(P_\mathcal{G}\overline{\mathbf{x}}) = P_\mathcal{G}\overline{f}(\overline{\mathbf{x}}) = PP_\mathcal{Q}\overline{f}(\overline{\mathbf{x}})$ for some $\overline{f}$. Therefore, $Pg(P_\mathcal{Q}\overline{\mathbf{x}}) = PP_\mathcal{Q}\overline{f}(\overline{\mathbf{x}})$. Since P always has full column rank, it is left-invertible, which means that $g(P_\mathcal{Q}\overline{\mathbf{x}}) = P_\mathcal{Q}\overline{f}(\overline{\mathbf{x}})$. That is, $\mathcal{A}_\mathcal{Q}$ is g-invariant for any $g \in F_\mathcal{Q}$, from which we conclude that $\mathcal{A}_\mathcal{G} \in L_{F_\mathcal{G}}$ implies $\mathcal{A}_\mathcal{Q} \in L_{F_\mathcal{Q}}$. We now prove the converse direction.
Assume $\mathcal{A}_\mathcal{Q} \in L_{F_\mathcal{Q}}$. Note that for any $f \in F_\mathcal{G}$, its quotient $g = f/\bowtie$ is in $F_\mathcal{Q}$. Then, from the fact that $\mathcal{A}_\mathcal{Q} \in L_{F_\mathcal{Q}}$ we know that $g(P_\mathcal{Q}\overline{\mathbf{x}}) = P_\mathcal{Q}\overline{g}(\overline{\mathbf{x}})$ for some $\overline{g}$. Multiplying on the left by P gives us $Pg(P_\mathcal{Q}\overline{\mathbf{x}}) = PP_\mathcal{Q}\overline{g}(\overline{\mathbf{x}}) = P_\mathcal{G}\overline{g}(\overline{\mathbf{x}})$. On the other hand, from the fact that $\bowtie$ is balanced, we have from Definition 3.7.4 that $Pg(P_\mathcal{Q}\overline{\mathbf{x}}) = f(PP_\mathcal{Q}\overline{\mathbf{x}}) = f(P_\mathcal{G}\overline{\mathbf{x}})$. Therefore, $f(P_\mathcal{G}\overline{\mathbf{x}}) = P_\mathcal{G}\overline{g}(\overline{\mathbf{x}})$. That is, $\mathcal{A}_\mathcal{G}$ is f-invariant for any $f \in F_\mathcal{G}$, from which we conclude that $\mathcal{A}_\mathcal{Q} \in L_{F_\mathcal{Q}}$ implies $\mathcal{A}_\mathcal{G} \in L_{F_\mathcal{G}}$, which completes the proof. ■

Note that this is in agreement with Corollary 3.7.12 when we consider the particular case $F_\mathcal{G} = \mathcal{F}_\mathcal{G}$ and $F_\mathcal{Q} = \mathcal{F}_\mathcal{Q}$. The following is now immediate from Theorem 3.7.13 and Lemma 3.3.7.

Corollary 3.7.14. *Consider networks $\mathcal{G}$ and $\mathcal{Q}$ such that $\mathcal{Q} = \mathcal{G}/\bowtie$ for some $\bowtie \in \Lambda_\mathcal{G}$, and subsets $F_\mathcal{G} \subseteq \mathcal{F}_\mathcal{G}$, $F_\mathcal{Q} \subseteq \mathcal{F}_\mathcal{Q}$ such that $F_\mathcal{Q} = F_\mathcal{G}/\bowtie$.*
Then, for $\mathcal{A} \leq \mathcal{T}_\mathcal{G}$ such that $\mathcal{A} \geq \bowtie$, we have that

$$cir_{F_\mathcal{G}}(\mathcal{A})/\bowtie = cir_{F_\mathcal{Q}}(\mathcal{A}/\bowtie). \tag{3.26}$$

□

3.8 CIR algorithm for balanced partitions

In this section we describe our improvement of the **CIR** algorithm that works with general weight sets and has a worst-case complexity of $\mathbf{O}(|\mathcal{C}|^3)$ in the case of a dense graph and $\mathbf{O}(|\mathcal{C}|^2)$ in the sparse case.
Consider a network represented by a matrix M together with an initial partition $\mathcal{A}_0 \leq \mathcal{T}$ represented by matrix P_0, of which we want to find the coarsest refinement (e.g., make P_0 the characteristic matrix of $\mathcal{T}$ if the goal is to find the maximal balanced partition $\top$).

3.8.1 Method

The idea of this algorithm is to start with the initial partition $\mathcal{A}_0$ and progressively refine it in a conservative manner. That is, given a partition $\mathcal{A}_i$, we construct a partition $\mathcal{A}_{i+1} \leq \mathcal{A}_i$ such that any balanced partition finer than $\mathcal{A}_i$ is also finer than $\mathcal{A}_{i+1}$. We create $\mathcal{A}_{i+1}$ by taking each color of $\mathcal{A}_i$ and splitting its cells according to whenever their corresponding rows in MP_i match or not. If $\mathcal{A}_{i+1} = \mathcal{A}_i$ the algorithm has converged and we found $\mathcal{A}_i = cir(\mathcal{A}_0)$, otherwise we continue iterating.

Lemma 3.8.1. *According to the described iterative method, any balanced partition finer than $\mathcal{A}_i$ is also finer than $\mathcal{A}_{i+1}$.* □

Proof. Assume that there are cells c, d such that $\mathcal{A}_i(c) = \mathcal{A}_i(d)$ but rows c and d of MP_i do not match perfectly (assume on k^{th} column). Note that the k^{th} color of $\mathcal{A}_i$ will correspond to either a color, or a union of colors of any balanced partition finer than $\mathcal{A}_i$. This means that no matter what refinement happens, the cells c and d will have no chance of having the same color in a balanced refinement, since if the sum of the parts is different, it will not be possible for the parts themselves to match. Therefore, any balanced partition finer than $\mathcal{A}_i$ is also finer than $\mathcal{A}_{i+1}$. ■

Remark 3.8.2. *Note that if at a certain iteration no more refinement happens, that means that the balanced condition Equation (3.16) has been achieved and we found $cir(\mathcal{A}_0)$.* □

Lemma 3.8.3. *The iterative procedure always converges in at most $|\mathcal{C}| - rank(\mathcal{A}_0)$ iterations.* □

Proof. Note that in each iteration, either the rank of the partition increases or the algorithm stops because a balanced partition was achieved. In the worst case scenario, the rank increases by one until the trivial partition is reached. Therefore, the algorithm always converges in at most $|\mathcal{C}| - rank(\mathcal{A}_0)$ iterations. ■

Since this algorithm always converges, this shows by construction that $cir(\mathcal{A}_0)$ exists. That is, for any partition $\mathcal{A}_0$, there is a unique balanced partition $\mathcal{A}_i = cir(\mathcal{A}_0)$ such $\mathcal{A}_i \leq \mathcal{A}_0$ and $\bowtie \leq \mathcal{A}_i$ for any balanced partition $\bowtie$ such that $\bowtie \leq \mathcal{A}_0$.

3.8.2 Efficient implementation and cost analysis

Note that a partition matrix P on a set of cells $\mathcal{C}$ can be efficiently represented by a vector of size $|\mathcal{C}|$ as seen in Example 3.1.7. Calculating the product MP_i consists on summing ($||$) certain elements of M according to the pattern described in P_i. To compare rows of MP_i previous works considered a quadratic cost which was the bottleneck of the algorithm. If the appropriate data structure (hash table) is used, such operation is of the order $\mathbf{O}(rank(\mathcal{A}_i))$. A pseudo-code description of the algorithm implementation is presented in Algorithms 1 and 2.

Algorithm 1 CIR algorithm

```
M ← CCN matrix
p_0 ← initial partition vector
r_0 ← rank of p_0
p_new ← p_0
r_new ← r_0
repeat
    p_old ← p_new
    r_old ← r_new
    (p_new, r_new) ← cir_iteration(M, p_old, r_old)
until r_new == r_old
```

Algorithm 2 CIR iteration

```
M ← CCN matrix
p_old ← previous partition vector
r_old ← rank of p_old
p_new ← new partition vector
r_new ← 0
for r = 1 : |C| do
    v ← zero vector of size r_old
    for c : (r, c) ∈ E do
        v[p_old(c)] ← v[p_old(c)] + M(r, c)
    end for
    s ← vec2string([p_old(r), v])
    value ← hash_table.find(s)
    if value NOT_FOUND then
        r_new ← r_new + 1
        p_new[r] ← r_new
        hash_table.insert(s, r_new)
    else
        p_new[r] ← value
    end if
end for
```

Lemma 3.8.4. *This implementation of the* ***CIR*** *algorithm leads to a worst-case complexity of* $\mathbf{O}(|\mathcal{C}|^3)$. □

Proof. In each iteration we are summing ($\|$) a total of $|\mathcal{E}|$ entries of M. The lookup and insertion in an hash table are fast operations with complexity $\mathbf{O}(1)$ which are each executed $|\mathcal{C}|$ times. The $|\mathcal{C}|$ strings that are used as key in the hash table have size proportional to $rank(\mathcal{A}_i)$.
The complexity of the i^{th} iteration is then $\mathbf{O}(|\mathcal{E}|+|\mathcal{C}|+|\mathcal{C}|rank(\mathcal{A}_i))$. In the worst-case scenario the rank increases by one and the number of iterations is $\mathbf{O}(|\mathcal{C}|)$. This implies total worst-case complexity of $\mathbf{O}(|\mathcal{C}|^3)$. ■

Remark 3.8.5. *In practice, the number of iterations seems to be much lower than $|\mathcal{C}|$ which means that this is a very pessimistic upper bound for the complexity.* □

We illustrate this algorithm with the following example.

Example 3.8.6. *Consider the network illustrated in Figure 3.8 with cell type partition $\mathcal{T} = \{\{1,2,5,6\},\{3,4\}\}$. The edge weight monoid is the same as in the parallel of resistors (Example 2.1.1). We assume that the arrows all represent values of* 30*. Note that the zero of the monoid is $0_{\mathcal{M}} = \infty$. This is represented by the matrix in Equation (3.27).*

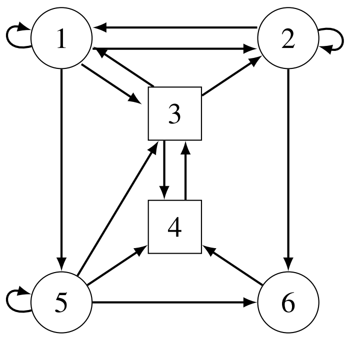


Figure 3.8: Network of Example 3.8.6 illustrating the CIR algorithm.

$$M = \begin{bmatrix} 30 & 30 & 30 & \infty & \infty & \infty \\ 30 & 30 & 30 & \infty & \infty & \infty \\ 30 & \infty & \infty & 30 & 30 & \infty \\ \infty & \infty & 30 & \infty & 30 & 30 \\ 30 & \infty & \infty & \infty & 30 & \infty \\ \infty & 30 & \infty & \infty & 30 & \infty \end{bmatrix}. \tag{3.27}$$

If we are interested in finding the top partition $\top$, we initialize $\mathcal{A}_0 = \mathcal{T}$. This partition can be represented by the matrix P_0

$$\mathcal{A}_0 = \begin{bmatrix} 1 \\ 1 \\ 2 \\ 2 \\ 1 \\ 1 \end{bmatrix}, \quad P_0 = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \\ 1 & 0 \end{bmatrix}.$$

Applying the algorithm we get

$$\left[\begin{array}{c|c} \mathcal{A}_0 & MP_0 \end{array}\right] = \left[\begin{array}{c|cc} 1 & 15 & 30 \\ 1 & 15 & 30 \\ 2 & 15 & 30 \\ 2 & 15 & 30 \\ 1 & 15 & \infty \\ 1 & 15 & \infty \end{array}\right]$$

whose row comparison determines the next iteration $\mathcal{A}_1$ and P_1

$$\mathcal{A}_1 = \begin{bmatrix} 1 \\ 1 \\ 2 \\ 2 \\ 3 \\ 3 \end{bmatrix}, \quad P_1 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}.$$

Applying the same procedure

$$\left[\begin{array}{c|c} \mathcal{A}_1 & MP_1 \end{array}\right] = \left[\begin{array}{c|ccc} 1 & 15 & 30 & \infty \\ 1 & 15 & 30 & \infty \\ 2 & 30 & 30 & 30 \\ 2 & \infty & 30 & 15 \\ 3 & 30 & \infty & 30 \\ 3 & 30 & \infty & 30 \end{array}\right]$$

and we get the second iteration defined by

$$\mathcal{A}_2 = \begin{bmatrix} 1 \\ 1 \\ 2 \\ 3 \\ 4 \\ 4 \end{bmatrix}, \quad P_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\left[\begin{array}{c|c} \mathcal{A}_2 & MP_2 \end{array}\right] = \left[\begin{array}{c|cccc} 1 & 15 & 30 & \infty & \infty \\ 1 & 15 & 30 & \infty & \infty \\ 2 & 30 & \infty & 30 & 30 \\ 3 & \infty & 30 & \infty & 15 \\ 4 & 30 & \infty & \infty & 30 \\ 4 & 30 & \infty & \infty & 30 \end{array}\right].$$

We can now see that $\mathcal{A}_2 = \mathcal{A}_3$*. This means that we have converged and* $\mathcal{A}_2 = cir(\mathcal{A}_0) = cir(\mathcal{T}) = \top$.

This is not the only non-trivial balanced partition on this network. For example, with an initial partition $\mathcal{B}_0 = \{\{1,2,5\},\{3,4\},\{6\}\}$ *we find the other balanced partition* $\mathcal{B}_1 = cir(\mathcal{B}_0) = \{\{1,2\},\{3\},\{4\},\{5\},\{6\}\}$.

Note that we already knew that any other balanced partitions would have to be finer that $\top$*. Therefore we could have instead just verified if any of the partitions* $\{\{1\},\{2\},\{3\},\{4\},\{5,6\}\}$ *or* $\{\{1,2\},\{3\},\{4\},\{5\},\{6\}\}$ *were balanced.* $\square$

Chapter 4

In-reachability based classification of synchrony partitions

In this chapter, we analyze the influence that the connectivity structure of a network has on its dynamics, in particular, with respect to the different types of (cumulative) in-neighborhoods and the in-reachability sets. This motivates the introduction of a qualitative classification scheme for the study of invariant synchrony sets.

4.1 Network connectivity

In this section we summarize the definitions and notation necessary to study the connectivity of a directed network and relate those characteristics to its dynamics. For an overview of the notions of in-neighborhoods, in-reachability and strongly connected components see Bang-Jensen and Gutin (2008).

4.1.1 Neighborhoods and reachability

Definition 4.1.1. *The* ***in-neighborhood*** $\mathcal{N}^-(c)$ *of a cell* $c \in \mathcal{C}$*, is the subset of cells* $d \in \mathcal{C}$ *such that the total of directed edges from* d *to* c *has a non-zero weight. Similarly, its out-neighborhood, denoted* $\mathcal{N}^+(c)$*, is the subset of cells* $d \in \mathcal{C}$ *such that the total of directed edges from* c *to* d *has a non-zero weight.* □

In our context, this means that if M is an in-adjacency matrix of a network, we have that $\mathcal{N}^-(c) = \{d \in \mathcal{C} \colon m_{cd} \neq 0_{ij}, i = \mathcal{T}(c), j = \mathcal{T}(d)\}$. Note that the commutative monoid structure allows us to encode arbitrary (finite) edges from a cell d to a cell c using a single element. This definition says that even if there are non-zero edges from d to c, if their total effect is equivalent to a non-edge (0_{ij}), then d is not in $\mathcal{N}^-(c)$.

Remark 4.1.2. *We often denote* $c \in \mathcal{N}^-(d)$*, or equivalently,* $d \in \mathcal{N}^+(c)$*, by* $c \to d$*.* □

Definition 4.1.3. *The* ***cumulative in-neighborhood*** $\mathcal{V}^-(c)$ *of a cell* $c \in \mathcal{C}$*, is defined as* $\mathcal{V}^-(c) := c \cup \mathcal{N}^-(c)$*.* □

Definition 4.1.4. *The k^{th} cumulative in-neighborhood $\mathcal{V}_k^-(c)$ of a cell $c \in \mathcal{C}$, is defined recursively as*

$$\mathcal{V}_0^-(c) := c, \tag{4.1}$$

$$\mathcal{V}_k^-(c) := \bigcup_{d \in \mathcal{V}_{k-1}^-(c)} \mathcal{V}^-(d), \quad k > 0. \tag{4.2}$$

That is, the set of cells from which there is a directed path of at most k edges that ends at c. Note that $\mathcal{V}_1^- = \mathcal{V}^-$. The k^{th} cumulative out-neighborhood $\mathcal{V}_k^+$ is defined similarly by replacing the signs. □

Lemma 4.1.5. *The sequence $\left(\mathcal{V}_k^-\right)_{k \geq 0}$ is monotonically increasing, that is,*

$$\mathcal{V}_k^-(c) \subseteq \mathcal{V}_{k+1}^-(c), \quad k \geq 0.$$

Moreover, if $\mathcal{V}_k^-(c) = \mathcal{V}_{k+1}^-(c)$ for some $k \geq 0$, then the recursion in Equation (4.2) has reached a fixed point, which means that $\mathcal{V}_k^-(c) = \mathcal{V}_n^-(c)$ for all $n \geq k$. □

This result motivates the following definition.

Definition 4.1.6. *The **in-reachability** $\mathcal{R}^-(c)$ of a cell $c \in \mathcal{C}$, is defined as*

$$\mathcal{R}^-(c) := \bigcup_{k \geq 0} \mathcal{V}_k^-(c). \tag{4.3}$$

That is, the set of cells from which there is a finite directed path that ends at c.
The out-reachability $\mathcal{R}^+$ is defined similarly by replacing the signs. □

Remark 4.1.7. *We often denote $c \in \mathcal{R}^-(d)$, or equivalently, $d \in \mathcal{R}^+(c)$, by $c \rightsquigarrow d$, illustrating that there is a direct path starting at cell c and ending at cell d.* □

Corollary 4.1.8. *For any cell c we have that $\mathcal{V}_k^-(c) \subseteq \mathcal{R}^-(c)$ for all $k \geq 0$. Moreover, when considering a finite amount of cells, equality is achieved at some finite k.* □

Corollary 4.1.9. *If $c \in \mathcal{R}^-(d)$, then $\mathcal{R}^-(c) \subseteq \mathcal{R}^-(d)$. That is, if $c \rightsquigarrow d$, then, for every cell e such that $e \rightsquigarrow c$ we also have that $e \rightsquigarrow d$.* □

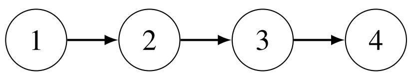


Figure 4.1: Simple chain of 4 cells.

Example 4.1.10. *Consider the simple network in Figure 4.1. Cell 3 receives an edge from cell 2, that is, $\mathcal{N}^-(3) = \{2\}$. Its cumulative in-neighborhood is given by $\mathcal{V}^-(3) = 3 \cup \mathcal{N}^-(3) = \{2,3\}$. Using the definition, its second cumulative in-neighborhood is $\mathcal{V}_2^-(3) = \mathcal{V}^-(2) \cup \mathcal{V}^-(3)$, which results in $\{1,2\} \cup \{2,3\} = \{1,2,3\}$, which are the cells that have a directed path to cell 3 with a*

length of two or less. Note that this is already the maximal cumulative in-neighborhood of cell 3 *since* $\mathcal{V}_3^-(3) = \mathcal{V}^-(1) \cup \mathcal{V}^-(2) \cup \mathcal{V}^-(3)$ *again equals* $\{1,2,3\}$*. That is,* $\mathcal{V}_2^-(3) = \mathcal{R}^-(3)$*.*
Furthermore, the point at which the cumulative in-neighborhoods equals the in-reachability set depends on the particular cell of the network. For instance, we have that $\mathcal{V}_0^-(1) = \mathcal{R}^-(1) = \{1\}$ *and* $\mathcal{V}_3^-(4) = \mathcal{R}^-(4) = \{1,2,3,4\}$*.*
Finally, we have that $\mathcal{R}^-(1) \subset \mathcal{R}^-(2) \subset \mathcal{R}^-(3) \subset \mathcal{R}^-(4)$ *since each cell has a direct path to every cell that is identified with an higher number. In particular, the set inclusions are strict, that is, there are no two cells with the same in-reachability set. Note that this would require directed loops, that is,* $\mathcal{R}^-(c) = \mathcal{R}^-(d)$ *is equivalent to* $\mathcal{R}^-(c) \subseteq \mathcal{R}^-(d)$ *and* $\mathcal{R}^-(d) \subseteq \mathcal{R}^-(c)$*, which implies* $c \in \mathcal{R}^-(d)$ *and* $d \in \mathcal{R}^-(c)$*. That is,* $c \rightsquigarrow d$ *and* $d \rightsquigarrow c$*.* □

4.1.2 Dynamics from in-neighborhoods

Consider a network $\mathcal{G}$ and an $\mathcal{G}$-admissible state set $\mathbb{X}$ such that a state $\mathbf{x} \in \mathbb{X}$ evolves (either discretely or continuously) according to a $\mathcal{G}$-admissible function $f \in \mathcal{F}_{\mathcal{G}}$. That is,

$$\mathbf{x}^+/\dot{\mathbf{x}} = f(\mathbf{x}). \tag{4.4}$$

From the definition of admissibility, the component f_c of an $\mathcal{G}$-admissible function f is only dependent on the states associated with the cells in $\mathcal{V}^-(c)$. This allows us to relate the dynamics of the system to the neighborhoods of cells. We now show how $\mathcal{V}_k^-$ in particular is related to the evolution of an admissible system in both the discrete and continuous cases.

Theorem 4.1.11. *Consider a network that evolves discretely according to a function* $f \in \mathcal{F}_{\mathcal{G}}$*. Then,* $x_c[n], x_c[n+1], \ldots, x_c[n+k]$ *are fully determined by the set of states* $\{x_d[n]\}$*, with* $d \in \mathcal{V}_k^-(c)$*.* □

Proof. It is enough to just prove that $x_c[n+k]$ is fully determined, the rest comes directly from the monotonicity of $(\mathcal{V}_k^-)_{k\geq 0}$.
The proof is by induction. Assume this to be true for some $k \geq 0$. Then, $x_c[n+k+1]$ is fully determined by the set of states $\{x_d[n+1]\}$ with $d \in \mathcal{V}_k^-(c)$. From f being $\mathcal{G}$-admissible, the states $\{x_d[n+1]\}$ themselves are fully determined by $\{x_e[n]\}$ with $e \in \mathcal{V}_1^-(d)$ for each $d \in \mathcal{V}_k^-(c)$. This means that $x_c[n+k+1]$ is fully determined by the states $\{x_d[n]\}$ with $d \in \mathcal{V}_{k+1}^-(c)$, which proves the induction step. The base case $k = 0$ is trivial. ■

Theorem 4.1.12. *Consider a system that evolves continuously according to a function* $f \in \mathcal{F}_{\mathcal{G}}$*. Then, assuming sufficient differentiability, the derivatives up to* k^{th} *order at time* t*, that is,* $x_c(t)$*,* $\dot{x}_c(t), \ldots, x_c^{(k)}(t)$ *are fully determined by the set of states* $\{x_d(t)\}$*, with* $d \in \mathcal{V}_k^-(c)$*.* □

Proof. It is enough to just prove that $x_c^{(k)}(t)$ is fully determined, the rest comes directly from the monotonicity of $(\mathcal{V}_k^-)_{k\geq 0}$.
The proof is by induction. Assume this to be true for some $k \geq 0$. Then, there is a function g such

that

$$x_c^{(k)}(t) = g(\{x_d(t) \colon d \in \mathcal{V}_k^-(c)\}).$$

Differentiating on both sides gives

$$x_c^{(k+1)}(t) = \sum_{d \in \mathcal{V}_k^-(c)} \frac{\partial g}{\partial x_d} x_d^{(1)}(t).$$

From f being $\mathcal{G}$-admissible, the first derivatives $\{x_d^{(1)}(t)\}$ are fully determined by $\{x_e(t)\}$ with $e \in \mathcal{V}_1^-(d)$ for each $d \in \mathcal{V}_k^-(c)$. This means that $x_c^{(k+1)}(t)$ is fully determined by the states $\{x_d(t)\}$ with $d \in \mathcal{V}_{k+1}^-(c)$. The base case $k = 0$ is trivial. ■

We now show that knowledge about the in-reachability $\mathcal{R}^-$ of a cell fully defines its evolution.

Theorem 4.1.13. *Consider a network that evolves either discretely or continuously, according to a function $f \in \mathcal{F}_{\mathcal{G}}$. Then, the whole $(x_c[k])_{k \geq n}/x_c(\cdot)$ is fully determined by the set of states $x_d[n]/x_d(t)$ for $d \in \mathcal{R}^-(c)$.* □

Proof. From Corollary 4.1.9, we know that for any in-reachability set $\mathcal{R}^-(c) = \mathcal{S}$, any cell $d \in \mathcal{S}$ has its own in-reachability contained within that same set. That is, $\mathcal{R}^-(d) \subseteq \mathcal{S}$. Since $\mathcal{V}^-(d) \subseteq \mathcal{R}^-(d)$, we have that $\mathcal{V}^-(d) \subseteq \mathcal{S}$.
From admissibility, we know that the dynamics of a cell d are a function of the states of the cells in $\mathcal{V}^-(d)$. Therefore, we can constrain our network to the subset of cells $\mathcal{S}$ while preserving all their dependencies within that same set. That is, knowledge about the initial conditions of the cells $\mathcal{S}$ is enough to fully determine the evolution of the induced subsystem. ■

Remark 4.1.14. *Note that for the discrete time case (Theorem 4.1.11), this result is direct from Corollary 4.1.8. However, to extend the continuous time case (Theorem 4.1.12) in the same manner, we would have to require the dynamics to be analytical, which is usually too much to ask for. Often, only the Lipschitz condition is assumed. Our approach in the previous proof works for both the discrete and continuous cases.* □

Corollary 4.1.15. *Consider a subset of cells $\mathcal{S}$ in a network that is an in-reachability set. That is, $\mathcal{S} = \mathcal{R}^-(c) \subseteq \mathcal{C}$ for some $c \in \mathcal{C}$. Then, for any solution $\mathbf{x}(t)$ of the whole system, constraining $\mathbf{x}(t)$ to the cells in $\mathcal{S}$ gives us a valid solution to the subnetwork induced by $\mathcal{S}$. Conversely, for a solution $\mathbf{x}_{\mathcal{S}}(t)$ on the subnetwork, there will be a solution on the whole network that is an extension of it.* □

Proof. This is direct from Theorem 4.1.13. ■

4.1.3 Strongly connected components and root dependency

To study the in-reachability sets $\mathcal{R}^-$ of the cells of the network, it is useful to decompose its graph into strongly connected components (**SCC**).

Definition 4.1.16. *Two cells $c, d \in \mathcal{C}$ are said to be **strongly connected** if $\mathcal{R}^-(c) = \mathcal{R}^-(d)$. That is, there are directed paths $d \rightsquigarrow c$ and $c \rightsquigarrow d$.* □

Remark 4.1.17. *Note that the strongly connected property induces a partition on the set of cells $\mathcal{C}$. The subsets of this partition are called the SCCs.* □

Since two cells in the same SCC have exactly the same in-reachability set, that is $\mathcal{R}^-(c) = \mathcal{R}^-(d)$ for all $c, d \in \mathcal{S}_i$, we simply refer to it as $\mathcal{R}^-(\mathcal{S}_i)$.

Definition 4.1.18. *The **condensation graph** is obtained by representing each SCC $\mathcal{S}_i$ by a block and connecting $\mathcal{S}_i \to \mathcal{S}_j$ for $i \neq j$, if there are cells $c_i \in \mathcal{S}_i$, $c_j \in \mathcal{S}_j$ such that $c_i \to c_j$.* □

The diagram obtained is **blockwise acyclic**. Note that for $c_i \in \mathcal{S}_i, c_j \in \mathcal{S}_j, i \neq j$, the existence of a directed path $c_i \rightsquigarrow c_j$ is equivalent to the existence of a directed path $\mathcal{S}_i \rightsquigarrow \mathcal{S}_j$ in the condensation graph. Moreover, if in the condensation graph there is a direct path $\mathcal{S}_i \rightsquigarrow \mathcal{S}_j$ then $\mathcal{S}_i \subseteq \mathcal{R}^-(\mathcal{S}_j)$. This decomposition can be done very efficiently in time $O(|\mathcal{C}| + |\mathcal{E}|)$, where $\mathcal{E}$ denotes the set of edges, using for instance Tarjan's algorithm Tarjan (1972).
Building on the concept of SCCs, we are now ready to define a decomposition based on root dependency components (**RDC**).

Definition 4.1.19. *An SCC $\mathcal{S}_i$ is called a **root** if there are no other SCCs that have a directed path to it. That is, $\mathcal{S}_i = \mathcal{R}^-(\mathcal{S}_i)$.* □

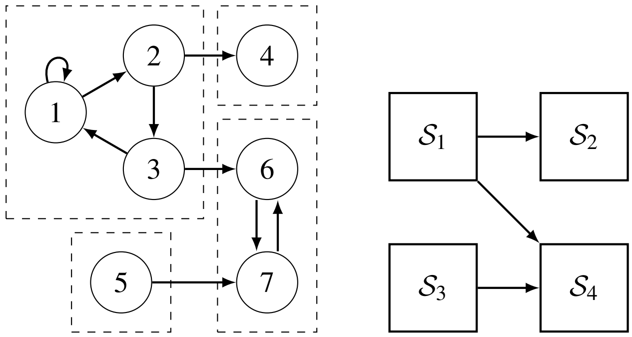
Definition 4.1.20. *Two cells $c, d \in \mathcal{C}$ are said to have the same **root dependency** if $\mathcal{R}^-(c)$, $\mathcal{R}^-(d)$ contain **exactly** the same subset of roots.* □

Remark 4.1.21. *Note that the property of having the same root dependency induces a partition on the set of cells $\mathcal{C}$. The subsets of this partition are called the **root dependency components**. Moreover, note that in network with n roots, this partition has at most $2^n - 1$ disjoint subsets, since there is no cell that does not depend on any root.* □

The following is straightforward from the definitions.

Corollary 4.1.22. *The partition formed by the SCCs is finer than the one formed by the RDCs.* □

Example 4.1.23. *Consider the network in Figure 4.2a. Note that it has four different SCCs. In particular, $\mathcal{S}_1 = \{1,2,3\}$, $\mathcal{S}_2 = \{4\}$, $\mathcal{S}_3 = \{5\}$ and $\mathcal{S}_4 = \{6,7\}$. This induces the partition $\{\{1,2,3\},\{4\},\{5\},\{6,7\}\}$ on the set of cells in the network. We form the condensation graph at Figure 4.2b by representing each SCC by a block and connecting them appropriately. That is, we have that $2 \to 4$, $3 \to 6$ and $5 \to 7$, which means that we need to connect $\mathcal{S}_1 \to \mathcal{S}_2$, $\mathcal{S}_1 \to \mathcal{S}_4$ and $\mathcal{S}_3 \to \mathcal{S}_4$, respectively.*



(a) Original network. (b) Condensation graph.

Figure 4.2: Decomposition of a network into its strongly connected components.

Using the condensation graph, it is very easy to see that the in-reachability sets of the SCCs are $\mathcal{R}^-(\mathcal{S}_1) = \mathcal{S}_1$, $\mathcal{R}^-(\mathcal{S}_2) = \mathcal{S}_1 \cup \mathcal{S}_2$, $\mathcal{R}^-(\mathcal{S}_3) = \mathcal{S}_3$ *and* $\mathcal{R}^-(\mathcal{S}_4) = \mathcal{S}_1 \cup \mathcal{S}_3 \cup \mathcal{S}_4$. *This means that the network has two roots,* $\mathcal{S}_1$ *and* $\mathcal{S}_3$. *With two roots, we can partition the cells of the network in, at most, three RDCs. That is, the ones that depend on the root* $\mathcal{S}_1$ *but not* $\mathcal{S}_3$ $(\mathcal{S}_1 \cup \mathcal{S}_2)$, *the ones that depend on* $\mathcal{S}_3$ *but not* $\mathcal{S}_1$ $(\mathcal{S}_3)$ *and the ones that depend on both* $\mathcal{S}_1$ *and* $\mathcal{S}_3$ $(\mathcal{S}_4)$. *Therefore, the partition induced by the RDCs is* $\{\mathcal{S}_1 \cup \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4\} = \{\{1,2,3,4\},\{5\},\{6,7\}\}$, *which is coarser that the partition of SCCs.* □

4.2 Strong, rooted and weak partitions

In this section, we classify the colors of partitions that represent synchrony patterns (as described in Chapter 3) according to their relationship to the structure of the network. To this purpose, we pay particular attention to the in-reachability sets, which fully determine the dynamical evolution of the cells, and the SCCs, which are the natural way of segmenting them.

Consider the network in Figure 4.2. Note that $\mathcal{S}_1$ and $\mathcal{S}_3$ are roots, that is, $\mathcal{R}^-(\mathcal{S}_1) = \mathcal{S}_1$ and $\mathcal{R}^-(\mathcal{S}_3) = \mathcal{S}_3$. From Theorem 4.1.13, the evolution of each of those sets can be completely determined without regard to the rest of the network. That is, for any $\mathcal{G}$-admissible function f, we can constraint and evaluate it separately in the sets of cells $\mathcal{S}_1$, $\mathcal{S}_3$.

Consider a partition $\mathcal{A}$ in this network such that there are cells in $\mathcal{S}_1$ and $\mathcal{S}_3$ that share the same color, that is, there are two cells $c_1 \in \mathcal{S}_1$, $c_3 \in \mathcal{S}_3$ such that $\mathcal{A}(c_1) = \mathcal{A}(c_3)$.

Since the two SCCs evolve completely decoupled from one another, any disturbance on c_1 would not be felt by c_3 and vice-versa. Moreover, there is no cell that could simultaneously affect both c_1 and c_3 and act as a pacemaker to drive them to a common state. However, this lack of feedback between these cells does not mean that it would be impossible for the synchrony pattern determined by $\mathcal{A}$ to appear in a physical system. That, is for states sufficiently close to the polydiagonal $\Delta_{\mathcal{A}}^{\mathbb{X}}$ to be driven back to $\Delta_{\mathcal{A}}^{\mathbb{X}}$, or at least stay close to it. This could be achieved if, for instance, both $x_{c_1}(t)$ and $x_{c_3}(t)$ converge to the same stable equilibrium point.

On the other hand, if $x_{c_1}(t), x_{c_3}(t)$ converge to the same limit cycle, we would not expect such synchrony space to be stable, since there would be no mechanism that could counteract a possible

phase offset. In particular, note that if $\mathbf{x}_{\mathcal{S}_1}(t)$ is a solution for the subnetwork induced by $\mathcal{S}_1$, the time shifted version $\mathbf{x}_{\mathcal{S}_1}(t-\delta)$ is also a solution. Therefore, phase synchronism with $\mathcal{S}_3$ would never happen unless we started with precise initial conditions.
Assume now that instead, there are two cells $c_2 \in \mathcal{S}_2$, $c_4 \in \mathcal{S}_4$ of the same color, that is $\mathcal{A}(c_2) = \mathcal{A}(c_4)$. Their in-reachability sets are $\mathcal{R}^-(\mathcal{S}_2) = \mathcal{S}_2 \cup \mathcal{S}_1$ and $\mathcal{R}^-(\mathcal{S}_4) = \mathcal{S}_4 \cup \mathcal{S}_1 \cup \mathcal{S}_3$ respectively. Now, although there is still no feedback between one another, their in-reachability sets intersect in $\mathcal{S}_1$. Thus, it could still be possible for c_2 and c_4 to maintain synchronism with non-trivial behavior if $\mathcal{S}_1$ is driving them to do so.
This shows that the structure of the network can make a crucial difference in the qualitative behavior of the invariant synchrony patterns, which motivates the following definitions.

Definition 4.2.1. *A color A on a partition of a network $\mathcal{G}$ is*

- ***Strong*** *if all the cells of that color are in the same SCC. That is,*

$$c,d \in A \implies \mathcal{R}^-(c) = \mathcal{R}^-(d), \tag{4.5}$$

  or equivalently,

$$\bigcap_{c\in A} \mathcal{R}^-(c) = \bigcup_{c\in A} \mathcal{R}^-(c). \tag{4.6}$$

- ***Rooted*** *if it is not strong but there is some cell (root) in $\mathcal{G}$ that has a directed path to all the cells of that color. That is,*

$$\emptyset \subset \bigcap_{c\in A} \mathcal{R}^-(c) \subset \bigcup_{c\in A} \mathcal{R}^-(c). \tag{4.7}$$

- ***Weak*** *if it is neither strong nor rooted. That is,*

$$\bigcap_{c\in A} \mathcal{R}^-(c) = \emptyset. \tag{4.8}$$

□

Clearly, every color on a partition is of one, and only one, of these three types. The following properties follow directly from the definition.

Lemma 4.2.2. *Consider a strong color A_s, a rooted color A_r, and a weak color A_w. Then, the following is true*

- *If $A \subseteq A_s$, then A is strong.*
- *If $A \subseteq A_r$, then A is either rooted or strong.*
- *If $A_r \subseteq A$, then A is either rooted or weak.*

- *If $A_w \subseteq A$, then A is weak.*

□

Note that Definition 4.2.1 classifies a particular color of some partition on $\mathcal{G}$ with respect to the connectivity structure of $\mathcal{G}$. This classification scheme is independent of the underlying partition containing that color. Furthermore, we do not assume any particular structure on the underlying partitions, such as being balanced or being finer that the partition of cell types $\mathcal{T}_{\mathcal{G}}$.
Using this classification scheme for individual colors, we classify a whole partition according to the following definition.

Definition 4.2.3. *A partition $\mathcal{A}$ on a network $\mathcal{G}$ is*

- ***Strong*** *if all of its colors are strong.*
- ***Rooted*** *if it is not strong but all of its colors are either rooted or strong. That is, it has at least one rooted color.*
- ***Weak*** *if any of its colors is weak.*

□

Clearly, every partition is of one, and only one, of these three types. Similarly to Lemma 4.2.2, the following properties are direct.

Lemma 4.2.4. *Consider a strong partition $\mathcal{A}_s$, a rooted partition $\mathcal{A}_r$ and a weak partition $\mathcal{A}_w$. Then, the following is true*

- *If $\mathcal{A} \leq \mathcal{A}_s$, then $\mathcal{A}$ is strong.*
- *If $\mathcal{A} \leq \mathcal{A}_r$, then $\mathcal{A}$ is either rooted or strong.*
- *If $\mathcal{A}_r \leq \mathcal{A}$, then $\mathcal{A}$ is either rooted or weak.*
- *If $\mathcal{A}_w \leq \mathcal{A}$, then $\mathcal{A}$ is weak.*

□

The following is straightforward.

Corollary 4.2.5. *If a color A is a singleton set, then is it strong. Furthermore, the trivial partition $\perp$, whose colors are all singleton sets is always strong.* □

We now relate our classification of partitions to the network connectivity according to the decomposition into SCCs and RDCs, as defined in Section 4.1.3. The two following results are direct from the definitions.

Lemma 4.2.6. *A partition is strong if and only if it is finer than the partition of SCCs.* □

We use the term **non-weak** to denote partitions or colors that are not weak, that is, either rooted or strong.

Lemma 4.2.7. *A partition finer than the partition of RDCs is non-weak.* □

In Section 3.5 we have seen that for any particular subset of functions $F \subseteq \mathcal{F}_{\mathcal{G}}$, the subset of partitions that are F-invariant always forms a lattice L_F. Furthermore, we know that its minimal element is always the trivial partition $\perp$, which is strong. Also, given any two partitions $\mathcal{A}_1, \mathcal{A}_2 \in L_F$, their least upper bound is always given by $\mathcal{A}_1 \vee \mathcal{A}_2$, where $\vee$ denotes the partition join operation as defined in Lemma 3.2.2. We now show how the join operation interacts with the proposed classification scheme.

Lemma 4.2.8. *For any pair of strong partitions $\mathcal{A}_1, \mathcal{A}_2$ on a network $\mathcal{G}$, their join $\mathcal{A} = \mathcal{A}_1 \vee \mathcal{A}_2$ is strong.* □

Proof. Since $\mathcal{A}_1, \mathcal{A}_2$ are strong, from Lemma 4.2.6, they are finer than the partition of SCCs. Then, $\mathcal{A} = \mathcal{A}_1 \vee \mathcal{A}_2$ is also finer than the partition of SCCs. From Lemma 4.2.6 again, $\mathcal{A}$ is strong. ■

This result, together with Lemma 4.2.4, allows us to understand how the join operation affects our connectivity-based classification of general partitions. This is summarized in Table 4.1, where S, R and W denote the partition classifications of strong, rooted and weak, respectively. So far we

$\vee$	S	R	W
S	S	R/W	W
R	R/W	R/W	W
W	W	W	W

Table 4.1: Join table for general partitions.

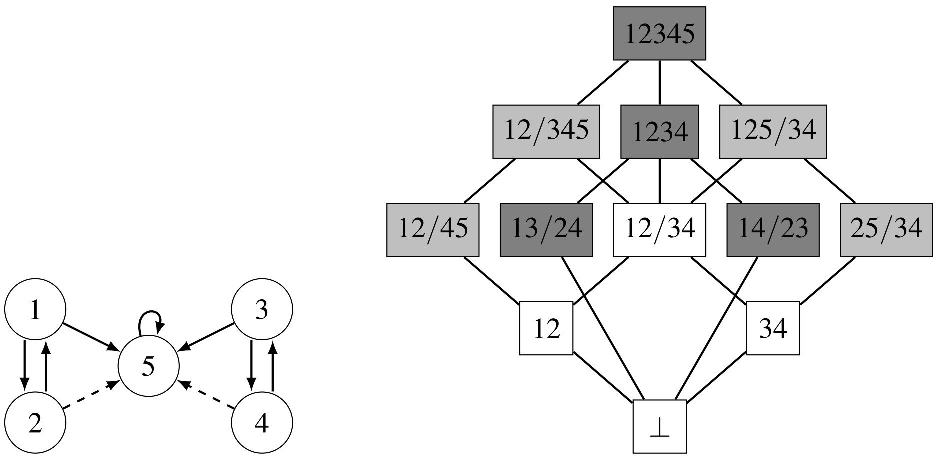
have not made any assumptions about the partitions. Moreover, we see in Table 4.1 that there are entries in which the classification is not completely defined. In particular, there are cases where the result of the join could be either rooted or weak (R/W).
Denote the subset of strong partitions in a lattice L_F by L_F^S and the subset of non-weak partitions by L_F^{NW}. Then, we have that

$$L_F^S \subseteq L_F^{NW} \subseteq L_F. \tag{4.9}$$

From Lemma 4.2.8, together with the fact that the trivial partition $\perp$ is strong, we know that L_F^S always forms a sublattice of L_F with a top element $\top_F^S$. On the other hand, L_F^{NW} might or might not be a lattice. This is illustrated in the following example.

Example 4.2.9. *Consider the network in Figure 4.3a and its respective lattice of balanced partitions Λ in Figure 4.3b. Consider the full edges to have a weight of 1 and the dashed edges to have weights of -1.*

In the lattice schematics, the partitions are colored according to their type such that strong partitions are in white, rooted ones are light gray and weak ones are in dark gray.
Note that Λ^S*, consisting of partitions in white, forms a sublattice of* Λ *with top partition* $\top^S = 12/34$*. On the other hand,* Λ^{NW} *does not form a lattice. In particular, if we join one of* $12/45$, $12/345$ *with one of* $25/34$, $125/34$*, we get* 12345*, which is a weak partition.* □



(a) Network.

(b) Lattice of balanced partitions.

Figure 4.3: A network and its lattice of balanced partitions.

From Lemma 4.2.4, we know that knowledge of the top partition $\top_F$ of a lattice L_F, with $F \subseteq \mathcal{F}_{\mathcal{G}}$, can give us important information about the whole lattice.

Corollary 4.2.10. *If the top partition* $\top_F$ *of a lattice* L_F*, with* $F \subseteq \mathcal{F}_{\mathcal{G}}$*, is non-weak, then all of its partitions are non-weak. Moreover, if* $\top_F$ *is strong, then all partitions are also strong.* □

We now show how the top strong partition $\top_F^S$ is given in terms of the cir_F function.

Corollary 4.2.11. *Consider a network* $\mathcal{G}$ *with cell type partition* $\mathcal{T}$*. Represent its SCCs according to a partition* $\mathcal{A}$*. Then,* $\top_F^S = cir_F(\mathcal{T} \wedge \mathcal{A})$*.* □

Note that L_F^{NW} is not necessarily a lattice and there might exist multiple locally maximal non-weak partitions, as in Example 4.2.9. In the following section we see that under some relatively tame assumptions, the resulting join table becomes much cleaner and we can guarantee that L_F^{NW} is a lattice with some top partition $\top_F^{NW}$.

4.3 Neighborhood color matching

In this section we present a sequence of progressively weaker assumptions about a partition on a network. We show that the weakest of them is enough to fix the remaining uncertain entries of Table 4.1 into Table 4.2.
We use the notation convention $\mathcal{A}(\mathbf{s}) := \bigcup_{c \in \mathbf{s}} \mathcal{A}(c)$. That is, $\mathcal{A}(\mathbf{s})$ denotes the subset of colors that are present in the set of cells $\mathbf{s} \subseteq \mathcal{C}$, according to the coloring assigned by $\mathcal{A}$.

Definition 4.3.1. *Consider a function $\mathcal{U}$ that assigns to each cell a subset of cells. That is, $\mathcal{U} \colon \mathcal{C} \to 2^{\mathcal{C}}$. Then, a partition $\mathcal{A}$ on $\mathcal{C}$ is $\mathcal{U}$-**matched** if when we apply $\mathcal{U}$ to cells of the same color, the resulting subsets share the exact same subset of colors. That is,*

$$\mathcal{A}(c) = \mathcal{A}(d) \implies \mathcal{A}(\mathcal{U}(c)) = \mathcal{A}(\mathcal{U}(d)). \tag{4.10}$$

□

Corollary 4.3.2. *The trivial partition $\perp$ is $\mathcal{U}$-matched for every function $\mathcal{U}$.* □

In this work, we are interested in the situation where the function $\mathcal{U}$ in Definition 4.3.1 denotes a neighborhood as described in Section 4.1.1, such as $\mathcal{N}^-$, $\mathcal{V}^-$, $\mathcal{V}_k^-$ or $\mathcal{R}^-$.

Corollary 4.3.3. *If a partition $\mathcal{A}$ is $\mathcal{N}^-$-matched, then, it is $\mathcal{V}^-$-matched.* □

Proof. If $\mathcal{A}(c) = \mathcal{A}(d)$, from assumption we have that $\mathcal{A}(\mathcal{N}^-(c)) = \mathcal{A}(\mathcal{N}^-(d))$. Then, we have that

$$\begin{aligned}
\mathcal{A}(c) \cup \mathcal{A}(\mathcal{N}^-(c)) &= \mathcal{A}(d) \cup \mathcal{A}(\mathcal{N}^-(d)) \\
\mathcal{A}(c \cup \mathcal{N}^-(c)) &= \mathcal{A}(d \cup \mathcal{N}^-(d)) \\
\mathcal{A}(\mathcal{V}^-(c)) &= \mathcal{A}(\mathcal{V}^-(d)).
\end{aligned}$$

■

Lemma 4.3.4. *If a partition $\mathcal{A}$ is $\mathcal{V}^-$-matched, then, it is $\mathcal{V}_k^-$-matched for every $k \geq 1$.* □

Proof. The proof is by induction. We assume that the statement applies to a given k. That is, $\mathcal{A}$ is both $\mathcal{V}^-$-matched and $\mathcal{V}_k^-$-matched. Then, we know that

$$\mathcal{A}(\mathcal{V}_{k+1}^-(c)) = \mathcal{A}\left(\bigcup_{c^\star \in \mathcal{V}_k^-(c)} \mathcal{V}^-(c^\star)\right) = \bigcup_{c^\star \in \mathcal{V}_k^-(c)} \mathcal{A}(\mathcal{V}^-(c^\star)),$$

where the first equality comes from Equation (4.2) and the second from how we defined the notation of applying $\mathcal{A}$ to a set. Since $\mathcal{A}$ is $\mathcal{V}^-$-matched, we know that $\mathcal{A}(\mathcal{V}^-(c^\star))$ only depends on the color of the cell $c^\star$. Moreover, since $\mathcal{A}$ is also $\mathcal{V}_k^-$-matched we know that $\mathcal{A}(c) = \mathcal{A}(d)$ implies $\mathcal{A}(\mathcal{V}_k^-(c)) = \mathcal{A}(\mathcal{V}_k^-(d))$, which means that $c^\star \in \mathcal{V}_k^-(c)$ and $d^\star \in \mathcal{V}_k^-(d)$ index the exact same set of colors. Therefore, $\mathcal{A}(\mathcal{V}_{k+1}^-(c)) = \mathcal{A}(\mathcal{V}_{k+1}^-(d))$.
The base case $k = 1$ is trivial since $\mathcal{V}_1^- = \mathcal{V}^-$. ■

Corollary 4.3.5. *If a partition defined on a finite set of cells is $\mathcal{V}^-$-matched, then it is also $\mathcal{R}^-$-matched.* □

Proof. This is direct from Lemma 4.3.4 and Corollary 4.1.8. ■

We have seen in Corollary 4.3.3 that a $\mathcal{N}^-$-matched partition is also $\mathcal{V}^-$-matched. The next very trivial example shows that the converse is not necessarily true.

Example 4.3.6. *Consider the network in Figure 4.4, which is colored with only one color (white). Note that $\mathcal{N}^-(1) = \{\}$ is empty and $\mathcal{N}^-(2) = \{1\}$ contains the white color. Therefore, the partition is not $\mathcal{N}^-$-matched. On the other hand, we have that $\mathcal{V}^-(1) = \{1\}$ and $\mathcal{V}^-(2) = \{1,2\}$ which means that the partition is $\mathcal{V}^-$-matched.* □

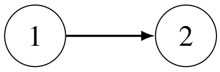


Figure 4.4: Partition that is not $\mathcal{N}^-$-matched but is $\mathcal{V}^-$-matched.

We have also seen in Corollary 4.3.5 that a $\mathcal{V}^-$-matched partition is also $\mathcal{R}^-$-matched. In the next example we disprove the converse statement.

Example 4.3.7. *Consider the network in Figure 4.5. Note that $\mathcal{V}^-(1) = \{1,4\}$ contains only white colors and $\mathcal{V}^-(4) = \{2,3,4\}$ contains white and gray colors. Therefore, the partition is not $\mathcal{V}^-$-matched. On the other hand, we have that $\mathcal{R}^-(1) = \mathcal{R}^-(4)$ and $\mathcal{R}^-(2) = \mathcal{R}^-(3)$, which means that the partition is $\mathcal{R}^-$-matched.* □

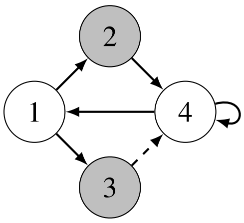


Figure 4.5: Partition that is not $\mathcal{V}^-$-matched but is $\mathcal{R}^-$-matched.

In summary, we have shown that the sequence: $\mathcal{N}^-$-matched , $\mathcal{V}^-$-matched and $\mathcal{R}^-$-matched lists progressively weaker assumptions. Note that in Example 4.3.7 the in-reachability sets are, in fact, all the same. The following result should be obvious from the definitions.

Corollary 4.3.8. *In a network that is a SCC, every partition is $\mathcal{R}^-$-matched.* □

More generally,

Corollary 4.3.9. *Every strong partition is $\mathcal{R}^-$-matched.* □

We now show that the tamest assumption we described ($\mathcal{R}^-$-matched) is enough to allow the following results.

Lemma 4.3.10. *If a non-weak partition is $\mathcal{R}^-$-matched, then it is finer than the partition of RDCs.* □

Proof. Consider some network with n roots $\mathcal{S}_1, \ldots, \mathcal{S}_n$ and a partition $\mathcal{A}$ that is non-weak and $\mathcal{R}^-$-matched. From the fact that they are roots, we have that $\mathcal{S}_i = \mathcal{R}^-(\mathcal{S}_i)$ and $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset$ for all $i \neq j$, which implies $\mathcal{R}^-(\mathcal{S}_i) \cap \mathcal{R}^-(\mathcal{S}_j) = \emptyset$ for all $i \neq j$. Then, since $\mathcal{A}$ is non-weak, $\mathcal{R}^-(\mathcal{S}_i) \cap \mathcal{R}^-(\mathcal{S}_j) = \emptyset$ implies $\mathcal{A}(\mathcal{S}_i) \cap \mathcal{A}(\mathcal{S}_j) = \emptyset$, for all $i \neq j$. That is, each root in the network contains a set of colors distinct from every other root.

Consider a cell c that is of a color that is present in one of the roots. That is, $\mathcal{A}(c) = k$ for some $k \in \mathcal{A}(\mathcal{S}_i)$. Then, since $\mathcal{A}$ is $\mathcal{R}^-$-matched, we have that $\mathcal{A}(\mathcal{R}^-(c)) = \mathcal{A}(\mathcal{S}_i)$. Since each root has a set of colors distinct from every other root, this means that $\mathcal{A}(\mathcal{R}^-(c)) \cap \mathcal{A}(\mathcal{S}_j) = \emptyset$ for all $j \neq i$. This implies $\mathcal{R}^-(c) \cap \mathcal{S}_j = \emptyset$ for all $j \neq i$. That is, if some cell in the network shares its color with a root, then that cell cannot depend (in the $\mathcal{R}^-$ sense) on any other roots. Since it is impossible to not depend on any roots at all, this implies that $\mathcal{R}^-(c) \supseteq \mathcal{S}_i$. That is, if a cell shares its color with a root, then it depends (in the $\mathcal{R}^-$ sense) on that root (and no other roots).

Finally, consider c, d such that $\mathcal{A}(c) = \mathcal{A}(d)$. Then, from $\mathcal{A}$ being $\mathcal{R}^-$-matched we have that $\mathcal{A}(\mathcal{R}^-(c)) = \mathcal{A}(\mathcal{R}^-(d))$. Since $\mathcal{R}^-(c)$ and $\mathcal{R}^-(d)$ share the exact same set of colors, they also share the same subset of colors that are present in roots. From what we have shown before, depending on a color shared by a root implies depending on the root itself. Therefore, cells of the same color depend on exactly the same roots, which means that $\mathcal{A}$ is finer than the partition of RDCs. ■

We now show how the top non-weak partition $\top_F^{NW}$ is given in terms of the cir_F function for the case where we know that all rooted partitions are $\mathcal{R}^-$-matched.

Corollary 4.3.11. *Consider a network $\mathcal{G}$ with cell type partition $\mathcal{T}$. Represent its RDCs according to a partition $\mathcal{B}$. Assume all its rooted partitions are $\mathcal{R}^-$-matched. Then, $\top_F^{NW} = cir_F(\mathcal{T} \wedge \mathcal{B})$.* □

We are now ready to prove the following result.

Lemma 4.3.12. *For any pair of non-weak $\mathcal{R}^-$-matched partitions $\mathcal{A}_{nw_1}, \mathcal{A}_{nw_2}$, their join $\mathcal{A}_{nw} = \mathcal{A}_{nw_1} \vee \mathcal{A}_{nw_2}$ is also non-weak.* □

Proof. From Lemma 4.3.10 we know that $\mathcal{A}_{nw_1}, \mathcal{A}_{nw_2}$ are both finer than the partition of RDCs. Therefore, their join $\mathcal{A}_{nw}$ is also going to be finer. From Lemma 4.2.7 we know that it is also non-weak. ■

The following result is straightforward from the general case illustrated in Table 4.1, together with Lemma 4.3.12.

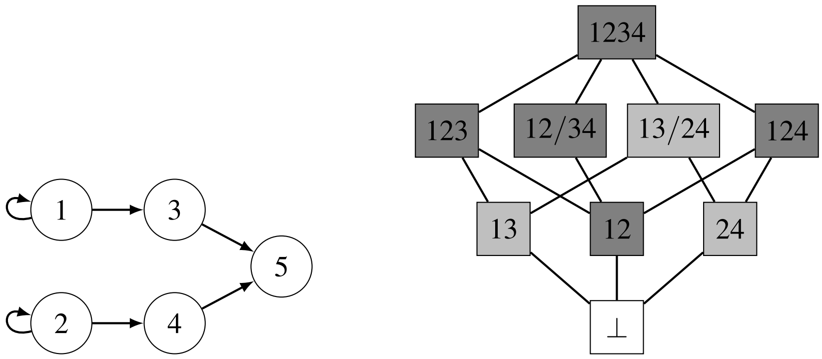
Corollary 4.3.13. *Consider partitions $\mathcal{A}_s, \mathcal{A}_{r_1}, \mathcal{A}_{r_2}$ such that $\mathcal{A}_s$ is strong and $\mathcal{A}_{r_1}, \mathcal{A}_{r_2}$ are rooted and $\mathcal{R}^-$-matched. Then, $\mathcal{A}_s \vee \mathcal{A}_{r_1}$ and $\mathcal{A}_{r_1} \vee \mathcal{A}_{r_2}$ are rooted.* □

This means that for the case where rooted partitions are $\mathcal{R}^-$-matched, Table 4.1 simplifies into Table 4.2. Furthermore, under such conditions we know that L_F^{NW} is a sublattice of L_F. This is illustrated in the following example.

Example 4.3.14. *Consider the network in Figure 4.6a and its respective lattice of balanced partitions Λ in Figure 4.6b. Note that Λ^S and Λ^{NW} are both lattices with top partitions $\top^S = \bot$ and $\top^{NW} = 13/24$, respectively. Note that every balanced partition in this network is $\mathcal{R}^-$-matched, therefore Table 4.2 applies. In the following section we will see that this fact is immediate from the network not allowing edge cancelings.* □

$\vee$	S	R	W
S	S	R	W
R	R	R	W
W	W	W	W

Table 4.2: Join table when rooted partitions are $\mathcal{R}^-$-matched.



(a) Network. (b) Lattice of balanced partitions.

Figure 4.6: A network and its lattice of balanced partitions.

4.4 Neighborhood color invariance

We now introduce a property that is stronger than Definition 4.3.1, that only applies to balanced partitions, since it is related to the respective quotient network.

Definition 4.4.1. *Consider a balanced partition $\bowtie \in \Lambda_{\mathcal{G}}$ on a network $\mathcal{G}$ and its respective quotient network $\mathcal{Q} = \mathcal{G}/\bowtie$. Take a particular type of neighborhood $\mathcal{U} \in \{\mathcal{N}^-, \mathcal{V}^-, \mathcal{V}_k^-, \mathcal{R}^-\}$ such that $\mathcal{U}_{\mathcal{G}}$ and $\mathcal{U}_{\mathcal{Q}}$ are the corresponding functions on $\mathcal{G}$ and $\mathcal{Q}$, respectively. Then, we say that $\bowtie$ is $\mathcal{U}$-**invariant** if*

$$d \in \mathcal{U}_{\mathcal{G}}(c) \implies \bowtie(d) \in \mathcal{U}_{\mathcal{Q}}(\bowtie(c)), \tag{4.11}$$

or equivalently,

$$\bowtie(\mathcal{U}_{\mathcal{G}}(c)) \subseteq \mathcal{U}_{\mathcal{Q}}(\bowtie(c)) \tag{4.12}$$

for all $c \in \mathcal{C}_{\mathcal{G}}$. □

We note that, as the following result shows, that the converse property of Definition 4.4.1 is always satisfied.

Lemma 4.4.2. *Consider a balanced partition $\bowtie \in \Lambda_{\mathcal{G}}$ on a network $\mathcal{G}$ and its respective quotient network $\mathcal{Q} = \mathcal{G}/\bowtie$. Then, for every color $A \in \bowtie$, which maps into the cell $k_A \in \mathcal{C}_{\mathcal{Q}}$, we have that*

$$k_A \in \mathcal{U}_{\mathcal{Q}}(\bowtie(c)) \implies A \cap \mathcal{U}_{\mathcal{G}}(c) \neq \emptyset, \tag{4.13}$$

or equivalently,

$$\mathcal{U}_{\mathcal{Q}}(\bowtie(c)) \subseteq \bowtie(\mathcal{U}_{\mathcal{G}}(c)) \tag{4.14}$$

for all $c \in \mathcal{C}_{\mathcal{G}}$. □

Proof. Firstly, we define $B \in \bowtie$ to be the color of c, mapping into the cell $k_B \in \mathcal{C}_{\mathcal{Q}}$.
Assume $k_A \in \mathcal{N}^-_{\mathcal{Q}}(k_B)$. Then, from the definition of $\mathcal{N}^-$, we have a non-zero entry $q_{k_B k_A} \neq 0_{ij}$, with $i = \mathcal{T}_{\mathcal{Q}}(k_B) = \mathcal{T}_{\mathcal{G}}(B)$ and $j = \mathcal{T}_{\mathcal{Q}}(k_A) = \mathcal{T}_{\mathcal{G}}(A)$ in the in-adjacency matrix Q associated with the quotient network $\mathcal{Q}$. Then, from the definition of quotient network, we have that $\sum_{d \in A \cap \mathcal{N}^-_{\mathcal{G}}(c)} w_{cd} = q_{k_B k_A}$. Since $q_{k_B k_A} \neq 0_{ij}$, this means that $A \cap \mathcal{N}^-_{\mathcal{G}}(c)$ is non-empty. That is, the statement is true for the case $\mathcal{U} = \mathcal{N}^-$.
We now prove the case $\mathcal{U} = \mathcal{V}^-$. Assume $k_A \in \mathcal{V}^-_{\mathcal{Q}}(\bowtie(c))$. Then, $k_A \in \{k_B \cup \mathcal{N}^-_{\mathcal{Q}}(k_B)\}$. Consider the case $k_A = k_B$. Then, $c \in A \cap \mathcal{V}^-_{\mathcal{G}}(c)$, which makes the set non-empty. Consider now the case $k_A \in \mathcal{N}^-_{\mathcal{Q}}(k_B)$. Then, since the statement is true for $\mathcal{U} = \mathcal{N}^-$, $A \cap \mathcal{N}^-_{\mathcal{G}}(c)$ is non-empty. Therefore, $A \cap \mathcal{V}^-_{\mathcal{G}}(c) \supseteq A \cap \mathcal{N}^-_{\mathcal{G}}(c)$ is also non-empty, which concludes the proof for $\mathcal{U} = \mathcal{V}^-$.
We now prove the case $\mathcal{U} = \mathcal{V}^-_k$ for every $k \geq 1$. The proof is by induction. Assume it to be true for a given k. Consider $k_A \in \mathcal{V}^-_{k+1\,\mathcal{Q}}(k_B)$. Then, $k_A \in \bigcup_{k_C \in \mathcal{V}^-_{k\,\mathcal{Q}}(k_B)} \mathcal{V}^-_{\mathcal{Q}}(k_C)$. That is, $k_A \in \mathcal{V}^-_{\mathcal{Q}}(k_C)$ for at least one particular $k_C \in \mathcal{V}^-_{k\,\mathcal{Q}}(k_B)$. Then, since the case $\mathcal{U} = \mathcal{V}^-_k$ is true from assumption, we have that $C \cap \mathcal{V}^-_{k\,\mathcal{G}}(c) \neq \emptyset$, where $C \in \bowtie$ is the color that maps into cell k_C. We choose a particular cell $d \in C \cap \mathcal{V}^-_{k\,\mathcal{G}}(c)$. Then, $\bowtie(d) = k_C$. Furthermore, since we know that the case $\mathcal{U} = \mathcal{V}^-$ is true, $k_A \in \mathcal{V}^-_{\mathcal{Q}}(k_C)$ implies $A \cap \mathcal{V}^-_{\mathcal{G}}(d) \neq \emptyset$. Finally, note that $A \cap \mathcal{V}^-_{k+1\,\mathcal{G}}(c) \supseteq A \cap \mathcal{V}^-_{\mathcal{G}}(d)$ since $d \in \mathcal{V}^-_{k\,\mathcal{G}}(c)$, which means that $A \cap \mathcal{V}^-_{k+1\,\mathcal{G}}(c)$ is also non-empty. This concludes the induction step. The base case $k = 1$ is trivial since $\mathcal{V}^-_1 = \mathcal{V}^-$.
Finally, the case $\mathcal{U} = \mathcal{R}^-$ is immediate from Corollary 4.1.8. ■

The following is immediate from Definition 4.4.1 and Lemma 4.4.2.

Corollary 4.4.3. *Consider a balanced partition* $\bowtie \in \Lambda_{\mathcal{G}}$ *on a network* $\mathcal{G}$ *and its respective quotient network* $\mathcal{Q} = \mathcal{G}/\bowtie$. *Then,* $\bowtie$ *is* $\mathcal{U}$*-invariant if and only if*

$$\bowtie(\mathcal{U}_{\mathcal{G}}(c)) = \mathcal{U}_{\mathcal{Q}}(\bowtie(c)) \tag{4.15}$$

for all $c \in \mathcal{C}_{\mathcal{G}}$. □

We now show that Definition 4.4.1 is a stronger property than the one in Definition 4.3.1.

Lemma 4.4.4. *If a balanced partition* $\bowtie$ *is* $\mathcal{U}$*-invariant, then, it is* $\mathcal{U}$*-matched.* □

Proof. Consider cells $c, d \in \mathcal{C}_{\mathcal{G}}$ in a network $\mathcal{G}$ such that $\bowtie(c) = \bowtie(d)$. Then, we have that $\mathcal{U}_{\mathcal{Q}}(\bowtie(c)) = \mathcal{U}_{\mathcal{Q}}(\bowtie(d))$ in the quotient network $\mathcal{Q} = \mathcal{G}/\bowtie$. Since $\bowtie$ is $\mathcal{U}$-invariant, from Corollary 4.4.3 we have that $\bowtie(\mathcal{U}_{\mathcal{G}}(c)) = \bowtie(\mathcal{U}_{\mathcal{G}}(d))$. Therefore, $\bowtie$ is $\mathcal{U}$-matched. ■

Lemma 4.4.5. *Consider a $\mathcal{U}$-invariant balanced partition $\bowtie \in \Lambda_{\mathcal{G}}$ on a network $\mathcal{G}$ and its respective quotient network $\mathcal{Q} = \mathcal{G}/\bowtie$. Then, for a partition $\mathcal{A} \geq \bowtie$, we have that*

$$\mathcal{A}(\mathcal{U}_{\mathcal{G}}(c)) = \mathcal{A}/\bowtie(\mathcal{U}_{\mathcal{Q}}(\bowtie(c))) \tag{4.16}$$

for all $c \in \mathcal{C}_{\mathcal{G}}$. □

Proof. From the fact that $\mathcal{A} \geq \bowtie$, we have that $\mathcal{A}(\mathcal{U}_{\mathcal{G}}(c)) = \mathcal{A}/\bowtie(\bowtie(\mathcal{U}_{\mathcal{G}}(c)))$. Since $\bowtie$ is $\mathcal{U}$-invariant, from Corollary 4.4.3, this becomes $\mathcal{A}/\bowtie(\mathcal{U}_{\mathcal{Q}}(\bowtie(c)))$. ■

Lemma 4.4.6. *Consider a $\mathcal{U}$-invariant balanced partition $\bowtie \in \Lambda_{\mathcal{G}}$ on a network $\mathcal{G}$ and its respective quotient network $\mathcal{Q} = \mathcal{G}/\bowtie$. Then, for a partition $\mathcal{A} \geq \bowtie$, we have that $\mathcal{A}$ is $\mathcal{U}$-matched in $\mathcal{G}$ if and only if $\mathcal{A}/\bowtie$ is $\mathcal{U}$-matched in $\mathcal{Q}$.* □

Proof. Firstly, note that $\mathcal{A}/\bowtie$ being $\mathcal{U}$-matched in $\mathcal{Q}$, from definition, means that $\mathcal{A}/\bowtie(\bowtie(c)) = \mathcal{A}/\bowtie(\bowtie(d))$ implies $\mathcal{A}/\bowtie(\mathcal{U}_{\mathcal{Q}}(\bowtie(c))) = \mathcal{A}/\bowtie(\mathcal{U}_{\mathcal{Q}}(\bowtie(d)))$. This simplifies into $\mathcal{A}(c) = \mathcal{A}(d)$ implies $\mathcal{A}/\bowtie(\mathcal{U}_{\mathcal{Q}}(\bowtie(c))) = \mathcal{A}/\bowtie(\mathcal{U}_{\mathcal{Q}}(\bowtie(d)))$. Therefore, we have to prove that if $\mathcal{A}(c) = \mathcal{A}(d)$, then $\mathcal{A}(\mathcal{U}_{\mathcal{G}}(c)) = \mathcal{A}(\mathcal{U}_{\mathcal{G}}(d))$ is equivalent to $\mathcal{A}/\bowtie(\mathcal{U}_{\mathcal{Q}}(\bowtie(c))) = \mathcal{A}/\bowtie(\mathcal{U}_{\mathcal{Q}}(\bowtie(d)))$. Since $\bowtie$ is $\mathcal{U}$-invariant, this is immediate from Lemma 4.4.5. ■

Lemma 4.4.7. *Consider a $\mathcal{U}$-invariant balanced partition $\bowtie_{01} \in \Lambda_{\mathcal{G}}$ on a network $\mathcal{G}$ and its respective quotient network $\mathcal{Q}_1 = \mathcal{G}/\bowtie_{01}$. Then, for a balanced partition $\bowtie_{02} \geq \bowtie_{01}$, we have that $\bowtie_{02}$ is $\mathcal{U}$-invariant in $\mathcal{G}$ if and only if $\bowtie_{12} := \bowtie_{02}/\bowtie_{01}$ is $\mathcal{U}$-invariant in $\mathcal{Q}_1$.* □

Proof. From Corollary 4.4.3, we have that $\bowtie_{02}$ being $\mathcal{U}$-invariant in $\mathcal{G}$ means that $\bowtie_{02}(\mathcal{U}_{\mathcal{G}}(c)) = \mathcal{U}_{\mathcal{Q}_2}(\bowtie_{02}(c))$, with $\mathcal{Q}_2 := \mathcal{G}/\bowtie_{02}$, for all $c \in \mathcal{C}_{\mathcal{G}}$. This can be rewritten, using $\bowtie_{12}$ as $\bowtie_{12}(\bowtie_{01}(\mathcal{U}_{\mathcal{G}}(c))) = \mathcal{U}_{\mathcal{Q}_2}(\bowtie_{12}(\bowtie_{01}(c)))$. Since from assumption, $\bowtie_{01}$ is $\mathcal{U}$-invariant, this can be equivalently written as $\bowtie_{12}(\mathcal{U}_{\mathcal{Q}_1}(\bowtie_{01}(c)) = \mathcal{U}_{\mathcal{Q}_2}(\bowtie_{12}(\bowtie_{01}(c)))$, for all $c \in \mathcal{C}_{\mathcal{G}}$. Using the mapping $d = \bowtie_{01}(c)$, it is easy to see that this is equivalent to $\bowtie_{12}(\mathcal{U}_{\mathcal{Q}_1}(d) = \mathcal{U}_{\mathcal{Q}_2}(\bowtie_{12}(d))$, for all $d \in \mathcal{C}_{\mathcal{Q}_1}$. Since from Lemma 3.7.3 we know that $\mathcal{Q}_2 = \mathcal{Q}_1/\bowtie_{12}$, this is equivalent to $\bowtie_{12}$ being $\mathcal{U}$-invariant in $\mathcal{Q}_1$. ■

Similarly to the $\mathcal{U}$-matched case, we have the following results.

Corollary 4.4.8. *The trivial partition $\perp$ is $\mathcal{U}$-invariant for every $\mathcal{U} \in \{\mathcal{N}^-, \mathcal{V}^-, \mathcal{V}_k^-, \mathcal{R}^-\}$.* □

Corollary 4.4.9. *If a balanced partition $\bowtie$ is $\mathcal{N}^-$-invariant, then, it is $\mathcal{V}^-$-invariant.* □

Proof. Consider cells $c, d \in \mathcal{C}_{\mathcal{G}}$ in a network $\mathcal{G}$ such that $c \in \mathcal{V}_{\mathcal{G}}^-(d)$. Then, $c \in d \cup \mathcal{N}_{\mathcal{G}}^-(d)$. Consider the case $c = d$. Then, $\bowtie(c) \in \mathcal{V}_{\mathcal{Q}}^-(\bowtie(c))$ in the quotient network $\mathcal{Q} = \mathcal{G}/\bowtie$, is immediate from the definition of $\mathcal{V}^-$. Consider now that $c \in \mathcal{N}_{\mathcal{G}}^-(d)$. Then, from $\bowtie$ being $\mathcal{N}^-$-invariant, we have that $\bowtie(c) \in \mathcal{N}_{\mathcal{Q}}^-(\bowtie(d))$, which implies $\bowtie(c) \in \mathcal{V}_{\mathcal{Q}}^-(\bowtie(d))$. ■

Lemma 4.4.10. *If a balanced partition $\bowtie$ is $\mathcal{V}^-$-invariant, then, it is $\mathcal{V}_k^-$-invariant for every $k \geq 1$.*
□

Proof. The proof is by induction. We assume that the statement applies to a given k. That is, $\bowtie$ is both $\mathcal{V}^-$-invariant and $\mathcal{V}_k^-$-invariant. Consider cells $c,d \in \mathcal{C}_\mathcal{G}$ in a network $\mathcal{G}$ such that $c \in \mathcal{V}^-_{k+1\,\mathcal{G}}(d)$. Then, $c \in \bigcup_{d^\star \in \mathcal{V}^-_{k\mathcal{G}}(d)} \mathcal{V}^-_\mathcal{G}(d^\star)$. That is, $c \in \mathcal{V}^-_\mathcal{G}(d^\star)$ for at least one particular $d^\star \in \mathcal{V}^-_{k\mathcal{G}}(d)$.

Then, from $\bowtie$ being $\mathcal{V}^-$-invariant and $\mathcal{V}_k^-$-invariant, we have that $\bowtie(c) \in \mathcal{V}^-_\mathcal{Q}(\bowtie(d^\star))$ and $\bowtie(d^\star) \in \mathcal{V}^-_{k\mathcal{Q}}(\bowtie(d))$, respectively, in the quotient network $\mathcal{Q} = \mathcal{G}/\bowtie$. Therefore, $\bowtie(c) \in \mathcal{V}^-_{k+1\,\mathcal{Q}}(\bowtie(d))$, which means that $\bowtie$ is $\mathcal{V}^-_{k+1}$-invariant.

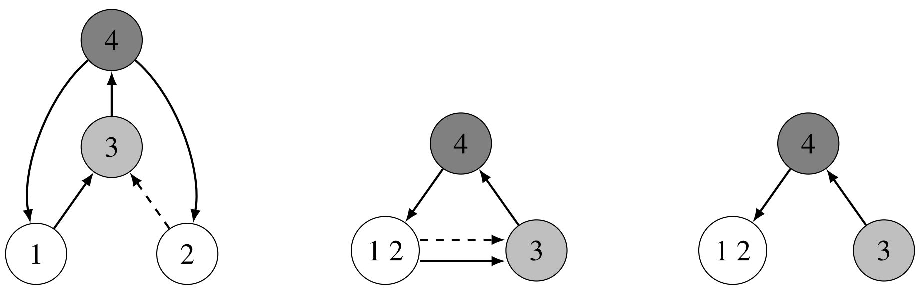
The base case $k=1$ is trivial since $\mathcal{V}_1^- = \mathcal{V}^-$. ■

Corollary 4.4.11. *If a balanced partition $\bowtie$ defined on a finite set of cells is $\mathcal{V}^-$-invariant, then it is also $\mathcal{R}^-$-invariant.* □

Proof. This is direct from Lemma 4.4.10 and Corollary 4.1.8. ■

We note that the concept of $\mathcal{U}$-invariance generalizes the concept of spurious partitions, which was defined in Aguiar et al. (2017). In particular, it corresponds to partitions not being $\mathcal{N}^-$-invariant. This is illustrated in the following example.

Example 4.4.12. *Consider the network in Figure 4.7a. Note that for a general admissible function $f \in \mathcal{F}_\mathcal{G}$, f_3 depends on the states of cells $1,2$. However, when the state is in $\Delta^{\mathbb{X}}_{\bowtie}$ with $\bowtie = \{\{1,2\},\{3\},\{4\}\}$, the total effect of cells $1,2$ on cell 3 cancels and 3 acts as if there were no edges coming from those cells.* □

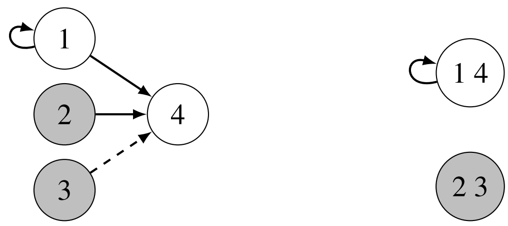


(a) Original network. (b) Edge canceling. (c) Quotient network.

Figure 4.7: Example of a spurious (not $\mathcal{N}^-$-invariant) partition.

Note that the partition in Example 4.4.12 is $\mathcal{N}^-$-matched despite not being $\mathcal{N}^-$-invariant. That is, while being $\mathcal{N}^-$-invariant is a sufficient condition for a partition to be $\mathcal{N}^-$-matched, it is not a necessary one. We now present an example that clarifies why the edge canceling in a balanced partition that is not $\mathcal{N}^-$-invariant might lead to it not being $\mathcal{N}^-$-matched.

Example 4.4.13. *Consider the network in Figure 4.8a, which is colored according to a balanced partition that is not $\mathcal{N}^-$-invariant (that is, it is spurious). Note that $\mathcal{N}^-(1) = \{1\}$ contains only white colors and $\mathcal{N}^-(4) = \{1,2,3\}$ contains white and gray colors. The fact that the edges coming from cells 2 and 3 cancel each other is exactly what allows this partition to be balanced despite this difference.* □

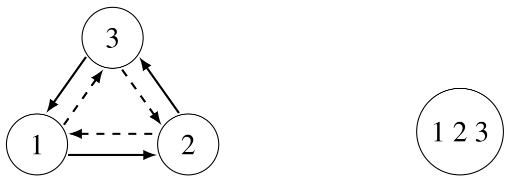


(a) Original network. (b) Quotient network.

Figure 4.8: Example of a partition that is neither $\mathcal{N}^-$-invariant nor $\mathcal{N}^-$-matched.

We have seen in Corollary 4.4.9 that a $\mathcal{N}^-$-invariant partition is also $\mathcal{V}^-$-invariant. The next example shows that the converse is not necessarily true.

Example 4.4.14. *Consider the network in Figure 4.9a, which is colored with a single color, according to the balanced partition $\{\{1,2,3\}\}$. Note that in this network, both the $\mathcal{N}^-$ and $\mathcal{V}^-$ in-neighborhoods of white cells contain white cells. On the other hand, in the quotient network in Figure 4.9b, we see that $\mathcal{N}^-$ of its only existing cell is empty. Therefore, this partition is not $\mathcal{N}^-$-invariant. It is, however, $\mathcal{V}^-$-invariant.* □



(a) Original network. (b) Quotient network.

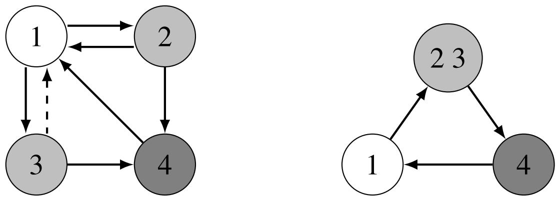
Figure 4.9: Example of a partition that is not $\mathcal{N}^-$-invariant but is $\mathcal{V}^-$-invariant.

We have also seen in Corollary 4.4.11 that a $\mathcal{V}^-$-invariant partition is also $\mathcal{R}^-$-invariant. In the next example we disprove the converse statement.

Example 4.4.15. *Consider the network in Figure 4.10a, which is colored according to the balanced partition $\{\{1\},\{2,3\},\{4\}\}$. Note that in the original network $\mathcal{G}$, we have that $\mathcal{V}^-(1) = \{1,2,3,4\}$. That is, white cells have white, light gray and dark gray colors in its $\mathcal{V}^-$ neighborhood. On the other hand, in the quotient, the white cell only has white and dark gray colors in its $\mathcal{V}^-$ neighborhood, which means that the partition is not $\mathcal{V}^-$-invariant. However, it is clear that the partition is $\mathcal{R}^-$-invariant, since in both the original network and in the quotient, all in-reachability sets $\mathcal{R}^-$ contain all the three colors of the partition.* □

In summary, we have shown that the sequence: $\mathcal{N}^-$-invariant , $\mathcal{V}^-$-invariant and $\mathcal{R}^-$-invariant lists progressively weaker assumptions.

Remark 4.4.16. *Refer back to Example 4.3.14. Note that the network only contains positive weights, which means that no matter which quotient we apply, there will be no edge canceling. That is, every balanced partition is immediately guaranteed to be $\mathcal{N}^-$-invariant. Then, it is also $\mathcal{R}^-$-invariant, which from Lemma 4.4.4 means that they are $\mathcal{R}^-$-matched.* □



(a) Original network. (b) Quotient network.

Figure 4.10: Example of a partition that is not $\mathcal{V}^-$-invariant but is $\mathcal{R}^-$-invariant.

We now show that the tamest assumption we defined in this section ($\mathcal{R}^-$-invariant) is enough to allow the following results.

Lemma 4.4.17. *Consider a $\mathcal{R}^-$-invariant balanced partition $\bowtie \in \Lambda_\mathcal{G}$ on a network $\mathcal{G}$ and its respective quotient network $\mathcal{Q} = \mathcal{G}/\bowtie$. If a partition $\mathcal{A} \geq \bowtie$ is strong in $\mathcal{G}$, then $\mathcal{A}/\bowtie$ is strong in $\mathcal{Q}$.* □

Proof. Firstly, note that $\mathcal{A}/\bowtie$ being strong in $\mathcal{Q}$, from definition, means that $\mathcal{A}/\bowtie(\bowtie(c)) = \mathcal{A}/\bowtie(\bowtie(d))$ implies $\mathcal{R}^-_\mathcal{Q}(\bowtie(c)) = \mathcal{R}^-_\mathcal{Q}(\bowtie(d))$. This simplifies into $\mathcal{A}(c) = \mathcal{A}(d)$ implies $\mathcal{R}^-_\mathcal{Q}(\bowtie(c)) = \mathcal{R}^-_\mathcal{Q}(\bowtie(d))$. Assume $\mathcal{A}(c) = \mathcal{A}(d)$. Then, from $\mathcal{A}$ being strong in $\mathcal{G}$, we have that $\mathcal{R}^-_\mathcal{G}(c) = \mathcal{R}^-_\mathcal{G}(d)$, which implies $\bowtie(\mathcal{R}^-_\mathcal{G}(c)) = \bowtie(\mathcal{R}^-_\mathcal{G}(d))$. Since $\bowtie$ is $\mathcal{R}^-$-invariant, we have from Corollary 4.4.3 that $\mathcal{R}^-_\mathcal{Q}(\bowtie(c)) = \mathcal{R}^-_\mathcal{Q}(\bowtie(d))$, which concludes the proof. ■

Lemma 4.4.18. *Consider a $\mathcal{R}^-$-invariant balanced partition $\bowtie \in \Lambda_\mathcal{G}$ on a network $\mathcal{G}$ and its respective quotient network $\mathcal{Q} = \mathcal{G}/\bowtie$. If a partition $\mathcal{A} \geq \bowtie$ is non-weak in $\mathcal{G}$, then $\mathcal{A}/\bowtie$ is non-weak in $\mathcal{Q}$.* □

Proof. Assume $\mathcal{A}$ is non-weak in $\mathcal{G}$. Then, for every color $A \in \mathcal{A}$, we have that $\bigcap_{c\in A} \mathcal{R}^-_\mathcal{G}(c) \neq \emptyset$. Then, we have that $\bowtie\left(\bigcap_{c\in A} \mathcal{R}^-_\mathcal{G}(c)\right) \neq \emptyset$. Note that $\bigcap_{c\in A} \bowtie(\mathcal{R}^-_\mathcal{G}(c)) \supseteq \bowtie\left(\bigcap_{c\in A} \mathcal{R}^-_\mathcal{G}(c)\right)$, therefore, $\bigcap_{c\in A} \bowtie(\mathcal{R}^-_\mathcal{G}(c)) \neq \emptyset$. Since $\bowtie$ is $\mathcal{R}^-$-invariant, we have from Corollary 4.4.3 that $\bigcap_{c\in A} \mathcal{R}^-_\mathcal{Q}(\bowtie(c)) \neq \emptyset$. This can written as $\bigcap_{\bowtie(c)\in A/\bowtie} \mathcal{R}^-_\mathcal{Q}(\bowtie(c)) \neq \emptyset$, which means that $\mathcal{A}/\bowtie$ is non-weak in $\mathcal{Q}$. ■

These results are summarized in the left hand side of Table 4.3, where, as before, S, R and W denote the partition classifications of strong, rooted and weak, respectively. The right hand side is easily seen to be equivalent to the left one. Note that for Table 4.3 to apply, we require the

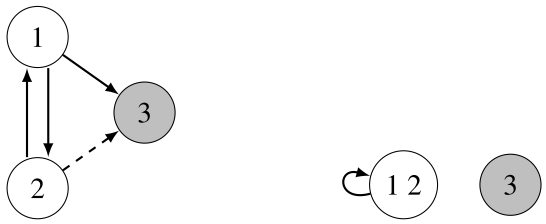
$\mathcal{A},\mathcal{G} \to$	$\mathcal{A}/\bowtie,\mathcal{Q}$
S	S
R	S/R
W	S/R/W

$\mathcal{A}/\bowtie,\mathcal{Q} \to$	$\mathcal{A},\mathcal{G}$
S	S/R/W
R	R/W
W	W

Table 4.3: Relation between partitions and their quotients over a $\mathcal{R}^-$-invariant partition.

partition we quotient over ($\bowtie$) to be $\mathcal{R}^-$-invariant. We now present some examples that show that these results do not apply if this assumption is not satisfied.

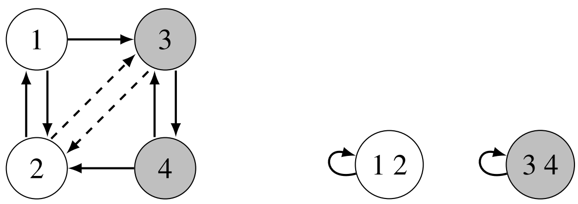
Example 4.4.19. *Consider the network $\mathcal{G}$ in Figure 4.11a, which is colored according to the balanced partition $\bowtie = \{\{1,2\},\{3\}\}$. Note that $\mathcal{R}^-_{\mathcal{G}}(3) = \{1,2,3\}$. That is, gray cells have white and gray colors in its $\mathcal{R}^-_{\mathcal{G}}$ neighborhood. On the other hand, in the quotient network $\mathcal{Q}$ in Figure 4.11b, we have that $\mathcal{R}^-_{\mathcal{Q}}(3) = \{3\}$. That is, the gray cell only has the gray color in its $\mathcal{R}^-_{\mathcal{Q}}$ neighborhood. Therefore, $\bowtie$ is not $\mathcal{R}^-$-invariant. Consider now the partition $\mathcal{A} = \{\{1,2,3\}\}$. Although this partition is rooted in $\mathcal{G}$, its quotient $\mathcal{A}/\bowtie = \{\{12,3\}\}$ is weak in $\mathcal{G}$.* □



(a) Original network. (b) Quotient network.

Figure 4.11: Example of a quotient over a partition that is not $\mathcal{R}^-$-invariant.

Example 4.4.20. *Consider the network $\mathcal{G}$ in Figure 4.12a, which is colored according to the balanced partition $\bowtie = \{\{1,2\},\{3,4\}\}$. Note that $\mathcal{G}$ consist of a single SCC. Therefore, each cell has white and gray colors in its $\mathcal{R}^-_{\mathcal{G}}$ neighborhood. On the other hand, in the quotient network $\mathcal{Q}$ in Figure 4.12b, we have that $\mathcal{R}^-_{\mathcal{Q}}(12) = \{12\}$. That is, the white cell only has the white color in its $\mathcal{R}^-_{\mathcal{Q}}$ neighborhood. Therefore, $\bowtie$ is not $\mathcal{R}^-$-invariant. Consider now the partition $\mathcal{A} = \{\{1,2,3,4\}\}$. Although this partition is strong in $\mathcal{G}$, its quotient $\mathcal{A}/\bowtie = \{\{12,34\}\}$ is weak in $\mathcal{G}$.* □



(a) Original network. (b) Quotient network.

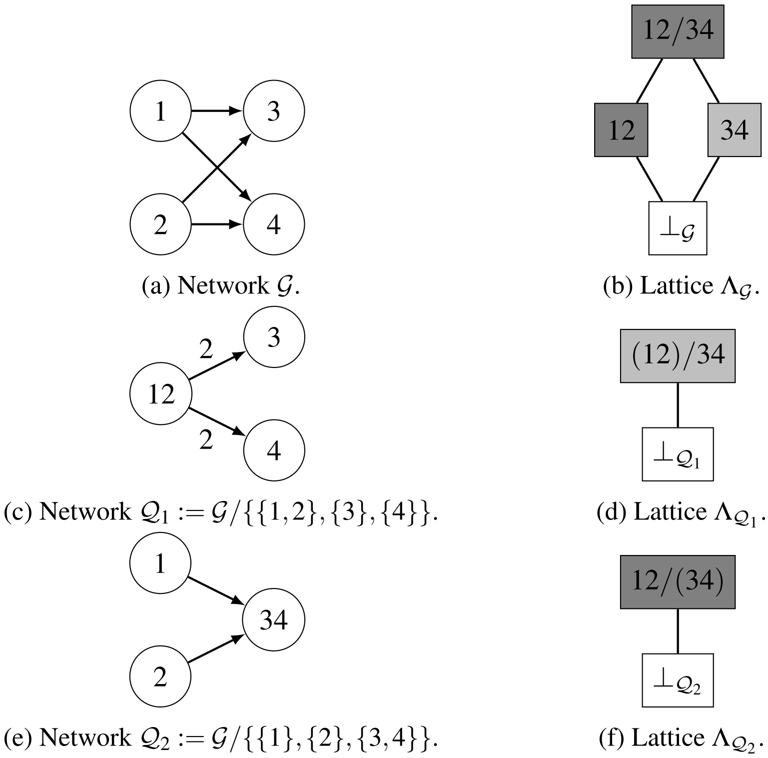
Figure 4.12: Example of a quotient over a partition that is not $\mathcal{R}^-$-invariant.

We now present examples where Table 4.3 does indeed apply.

Example 4.4.21. *Consider the network in Figure 4.13a and its respective lattice of balanced partitions $\Lambda_{\mathcal{G}}$ in Figure 4.13b. We define the quotient networks $\mathcal{Q}_1 := \mathcal{G}/\bowtie_1$, $\mathcal{Q}_2 := \mathcal{G}/\bowtie_2$ over the balanced partitions $\bowtie_1 = \{\{1,2\},\{3\},\{4\}\}$ and $\bowtie_2 = \{\{1\},\{2\},\{3,4\}\}$, respectively. Note that both $\bowtie_1$ and $\bowtie_2$ are $\mathcal{R}^-$-invariant, therefore, Table 4.3 applies.*

The set of partitions in $\Lambda_{\mathcal{G}}$ that are coarser than $\bowtie_1$ are $\{\{1,2\},\{3\},\{4\}\}$ ($\bowtie_1$ itself) and $\{\{1,2\},\{3,4\}\}$, which are both weak. In the lattice $\Lambda_{\mathcal{Q}_1}$ these two partitions correspond to $\bot_{\mathcal{Q}_1}$ and $\{\{12\},\{3,4\}\}$, which are strong and rooted, respectively.

The set of partitions in $\Lambda_{\mathcal{G}}$ that are coarser than $\bowtie_2$ are $\{\{1\},\{2\},\{3,4\}\}$ ($\bowtie_2$ itself) and $\{\{1,2\},\{3,4\}\}$, *which are rooted and weak respectively. In the lattice $\Lambda_{\mathcal{Q}_2}$ these two partitions correspond to $\bot_{\mathcal{Q}_2}$ and $\{\{1,2\},\{34\}\}$, which are strong and weak respectively.* $\square$



(a) Network $\mathcal{G}$. (b) Lattice $\Lambda_{\mathcal{G}}$.

(c) Network $\mathcal{Q}_1 := \mathcal{G}/\{\{1,2\},\{3\},\{4\}\}$. (d) Lattice $\Lambda_{\mathcal{Q}_1}$.

(e) Network $\mathcal{Q}_2 := \mathcal{G}/\{\{1\},\{2\},\{3,4\}\}$. (f) Lattice $\Lambda_{\mathcal{Q}_2}$.

Figure 4.13: Lattices of balanced partitions of a network and its quotients.

Note that we can always quotient a network over the trivial partition. That is, $\mathcal{Q} := \mathcal{G}/\bot_{\mathcal{G}}$. Consider we encode $\bot_{\mathcal{G}}$ through the identity mapping. In such case we have that $\mathcal{G} = \mathcal{Q}$ and $L_{F_{\mathcal{Q}}} = L_{F_{\mathcal{G}}}/\bot_{\mathcal{G}} = L_{F_{\mathcal{G}}}$. Therefore, every partition in $L_{F_{\mathcal{G}}}$ maps to itself in $L_{F_{\mathcal{Q}}}$. This implies the cases $S \to S$, $R \to R$ and $W \to W$ in the left side of Table 4.3.
On the other hand, for the case $\mathcal{Q} := \mathcal{G}/\bowtie$ and $L_{F_{\mathcal{Q}}} = L_{F_{\mathcal{G}}}/\bowtie$, for any $\bowtie \in \Lambda_{\mathcal{G}}$ we have that $\bowtie/\bowtie = \bot_{\mathcal{Q}}$. Since $\bot_{\mathcal{Q}}$ is always strong in $\mathcal{Q}$, this covers the cases $S/R/W \to S$ in the left side of Table 4.3. This means that most of the cases of Table 4.3 were forced. The remaining case $W \to R$, was illustrated in Example 4.4.21. That is, the interest of this result lies in the fact that it excludes most of the non-forced cases.

Chapter 5

Output vector spaces

In this chapter we present results that apply when the output sets $\{\mathbb{Y}_i\}_{i\in T}$ are vector spaces.

5.1 Admissible vector spaces and related results

In this section we present some direct consequences of the output spaces being vector spaces. Namely, the set of oracle functions $\hat{\mathcal{F}}_T$ and admissible functions $\mathcal{F}_{\mathcal{G}}$ also being vector spaces. Also, evaluation on a network $(|_{\mathcal{G}})$ is a linear map that maps $\hat{\mathcal{F}}_T$ into $\mathcal{F}_{\mathcal{G}}$. We also present some local robustness results related to invariant patterns and balanced partitions.

Lemma 5.1.1. *Consider $\hat{\mathcal{F}}_i$, defined on an output set $\mathbb{Y}_i$ that is a vector space. Then, $\hat{\mathcal{F}}_i$ is itself a vector space.* □

Proof. Consider components $\hat{f}_i, \hat{g}_i \in \hat{\mathcal{F}}_i$. Defining, $\hat{h}_i := \alpha \hat{f}_i + \hat{g}_i$ for some scalar α, we have that

$$\begin{aligned} \hat{h}_i(x;\mathbf{w},\mathbf{x}) &= \alpha \hat{f}_i(x;\mathbf{w},\mathbf{x}) + \hat{g}_i(x;\mathbf{w},\mathbf{x}) \\ &= \alpha \hat{f}_i(x;\sigma\mathbf{w},\sigma\mathbf{x}) + \hat{g}_i(x;\sigma\mathbf{w},\sigma\mathbf{x}) \\ &= \hat{h}_i(x;\sigma\mathbf{w},\sigma\mathbf{x}), \end{aligned}$$

which means that $\hat{h}_i$ satisfies item 1 of Definition 2.4.1. Items 2 and 3 are verified in the exact same way. Therefore $\hat{h}_i \in \hat{\mathcal{F}}_i$ and $\hat{\mathcal{F}}_i$ is a vector space. ■

Corollary 5.1.2. *Consider $\hat{\mathcal{F}}_T$, defined on the output sets $\{\mathbb{Y}_i\}_{i\in T}$ that are vector spaces. Then, $\hat{\mathcal{F}}_T$ is itself a vector space.* □

Lemma 5.1.3. *Assume $\hat{\mathcal{F}}_T$ is a vector space. Evaluation on a network $(|_{\mathcal{G}})$ is linear.* □

Proof. Consider oracle functions $\hat{f}, \hat{g} \in \hat{\mathcal{F}}_T$. Since $\hat{\mathcal{F}}_T$ is a vector space, there is a $\hat{h} \in \hat{\mathcal{F}}_T$ such that $\hat{h} = \alpha \hat{f} + \hat{g}$ for some scalar α. Define $f = \hat{f}|_{\mathcal{G}}$ and $g = \hat{g}|_{\mathcal{G}}$. Then $h = \hat{h}|_{\mathcal{G}}$ is such that

$$\begin{aligned} h_c(\mathbf{x}) &= \hat{h}_i(x_c; \mathbf{m}_c, \mathbf{x}) \\ &= \alpha \hat{f}_i(x_c; \mathbf{m}_c, \mathbf{x}) + \hat{g}_i(x_c; \mathbf{m}_c, \mathbf{x}) \\ &= \alpha f_c(\mathbf{x}) + g_c(\mathbf{x}) \end{aligned}$$

for all $c \in \mathcal{C}$ with $i = \mathcal{T}(c)$. That is, $(\alpha \hat{f} + \hat{g})|_{\mathcal{G}} = \alpha \hat{f}|_{\mathcal{G}} + \hat{g}|_{\mathcal{G}}$. Therefore, evaluation on a network $(|_{\mathcal{G}})$ is a linear map on $\hat{\mathcal{F}}_T$. ■

Corollary 5.1.4. *Consider some $\hat{F}_T \subseteq \hat{\mathcal{F}}_T$ and define $F_{\mathcal{G}} := \hat{F}_T|_{\mathcal{G}}$. If $\hat{F}_T$ is a vector space, then $F_{\mathcal{G}}$ is also a vector space.* □

Corollary 5.1.5. *Assume $\hat{\mathcal{F}}_T$ is a vector space. Then, evaluating at a network $\mathcal{G}$ $(|_{\mathcal{G}})$ partitions the space of functions $\hat{\mathcal{F}}_T$ into affine planes parallel to the kernel (or nullspace)* $\ker(|_{\mathcal{G}})$ *such that each plane represents the set of oracle functions that behave the same in that network, that is,*

$$\hat{f}|_{\mathcal{G}} = \hat{g}|_{\mathcal{G}} \iff \hat{f} - \hat{g} \in \ker(|_{\mathcal{G}})$$

for every $\hat{f}, \hat{g} \in \hat{\mathcal{F}}_T$. □

We now present synchrony properties for output vector spaces.

Lemma 5.1.6. *Consider some $F_{\mathcal{G}} \subseteq \mathcal{F}_{\mathcal{G}}$. Given some partition $\mathcal{A} \leq \mathcal{T}_{\mathcal{G}}$, define $F_{\mathcal{G}}^{\mathcal{A}} := \{f \in F_{\mathcal{G}} : \mathcal{A} \text{ is } f\text{-invariant}\}$. If $F_{\mathcal{G}}$ is a vector space, then $F_{\mathcal{G}}^{\mathcal{A}}$ is also a vector space.* □

Proof. Consider functions $f, g \in F_{\mathcal{G}}^{\mathcal{A}}$. Then, $f, g \in F_{\mathcal{G}}$. Defining, $h := \alpha f + g$ for some scalar α, we have that $h \in F_{\mathcal{G}}$ since, from assumption, $F_{\mathcal{G}}$ is a vector space. Furthermore, if $\mathbf{x} = P\overline{\mathbf{x}}$, where P is a partition matrix that represents $\mathcal{A}$, we have

$$\begin{aligned} h(P\overline{\mathbf{x}}) &= \alpha f(P\overline{\mathbf{x}}) + g(P\overline{\mathbf{x}}) \\ &= \alpha P\overline{f}(\overline{\mathbf{x}}) + P\overline{g}(\overline{\mathbf{x}}) \\ &= P\left(\alpha \overline{f}(\overline{\mathbf{x}}) + \overline{g}(\overline{\mathbf{x}})\right) \\ &= P\left(\overline{h}(\overline{\mathbf{x}})\right). \end{aligned}$$

Therefore $h \in F_{\mathcal{G}}^{\mathcal{A}}$, which means that $F_{\mathcal{G}}^{\mathcal{A}}$ is a vector space. ■

Corollary 5.1.7. *Consider some $\hat{F}_T \subseteq \hat{\mathcal{F}}_T$. Given some partition $\mathcal{A} \leq \mathcal{T}_{\mathcal{G}}$, define $F_{\mathcal{G}}^{\mathcal{A}} := \{f \in \hat{F}_T|_{\mathcal{G}} : \mathcal{A} \text{ is } f\text{-invariant}\}$. If $\hat{F}_T$ is a vector space, then $F_{\mathcal{G}}^{\mathcal{A}}$ is also a vector space.* □

In a practical application, the nominal admissible function f^* that we desire in theory might not be the one that is actually realized. This motivates the interest in having some sort of local robustness so functions f that are sufficiently close to f^* show similar properties.

Corollary 5.1.8. *Consider some $F_\mathcal{G} \subseteq \mathcal{F}_\mathcal{G}$ such that $F_\mathcal{G}$ is a **normed** vector space. Given a $f^* \in F_\mathcal{G}$, if we require that for some $\varepsilon > 0$, all the functions $f \in F_\mathcal{G}$ in the ball $\|f - f^*\| < \varepsilon$ are such that $f \in F_\mathcal{G}^\mathcal{A}$, then $F_\mathcal{G} = F_\mathcal{G}^\mathcal{A}$.* □

From Corollary 3.6.15 the following is now immediate.

Corollary 5.1.9. *Assume $\mathcal{F}_\mathcal{G}$ is a **normed** vector space. Given a $f^* \in \mathcal{F}_\mathcal{G}$, if we require that for some $\varepsilon > 0$, all the functions $f \in \mathcal{F}_\mathcal{G}$ in the ball $\|f - f^*\| < \varepsilon$ are such that $f \in \mathcal{F}_\mathcal{G}^\mathcal{A}$, then $\mathcal{A} \in \Lambda_\mathcal{G}$.* □

We now present special cases for particular monoids in which $\mathcal{A}$ being $F_\mathcal{G}$-invariant for some subsets $F_\mathcal{G} \subseteq \mathcal{F}_\mathcal{G}$ is enough to enforce $\mathcal{A}$ to be balanced.
The usual proof for Theorem 3.6.13 in the unweighted and scalar-weighted cases, uses functions that are linear in the weights. This approach, however, does not scale well to general weight sets. Note that the analogous in this framework is to consider functions that are **additive in the weights**, that is,

$$p(w_1 \| w_2) = p(w_1) + p(w_2).$$

If there is an annihilator in $\mathcal{M}$, then $p(w) = 0_\mathcal{M}$ for all $w \in \mathcal{M}$. That is, only the trivial case for such functions exists. We now present an extension of the linear in the weights argument to a particular type of weight monoids for which it works.

Theorem 5.1.10. *Consider non-trivial output vector spaces $\{\mathbb{Y}_i\}_{i \in T}$ and assume that the edges are in the monoid $\mathcal{M} = \langle \mathbb{M} | E \rangle$, with $\mathbb{M} = \mathbb{R} \times \mathbb{W}$ and*

$$E = \{\lambda_1 w \| \lambda_2 w = (\lambda_1 + \lambda_2) w, \quad \forall \lambda_1, \lambda_2 \in \mathbb{R}, w \in \mathbb{W}\},$$

where $\mathbb{W}$ is not necessarily countable.
Consider the set of oracle components $\hat{f}_i$, $i \in T$, that are only dependent on neighbors that are in a specific state $\overline{x}_k$, such that

$$\hat{f}_i\left(x; \sum \lambda_w w, \overline{x}_k\right) = \lambda_e \mathbf{v}, \quad \mathbf{v} \in \mathbb{Y}_i, \mathbf{v} \neq 0_{\mathbb{Y}_i}$$

for some $e \in \mathbb{W}$.
If a partition $\mathcal{A}$ is invariant under the subset $F_\mathcal{G} \subseteq \mathcal{F}_\mathcal{G}$ that is constructed with oracle components $\hat{f}_i$ of the type above, then $\mathcal{A}$ is balanced in $\mathcal{G}$. □

Proof. We prove this through its contrapositive. That is, $\mathcal{A} \notin \Lambda_\mathcal{G}$ implies $\mathcal{A} \notin L_{F_\mathcal{G}}$.
Consider any partition $\mathcal{A} \leq \mathcal{T}$ that is not balanced. Then, there are cells $c, d \in \mathcal{C}$ such that $\mathcal{A}(c) = \mathcal{A}(d)$ and $P^\top \mathbf{m}_c^\top \neq P^\top \mathbf{m}_d^\top$, where P is a partition matrix of $\mathcal{A}$. Consider k one of the entries in which they differ. That is, $\left[P^\top \mathbf{m}_c^\top\right]_k \neq \left[P^\top \mathbf{m}_d^\top\right]_k$.
We now choose a state in the related polydiagonal $\mathbf{x} \in \Delta_\mathcal{A}^\mathbb{X}$, that is, $\mathbf{x} = P\overline{\mathbf{x}}$ for some $\overline{\mathbf{x}}$, such that $\overline{x}_k$ is different from all other entries of $\overline{\mathbf{x}}$.

An element of the monoid $\mathcal{M}$ can be written as linear combination over a finite subset of elements in $\mathbb{W}$, that is, $\sum \lambda_w w$. If $\left[P^\top \mathbf{m}_c^\top\right]_k \neq \left[P^\top \mathbf{m}_d^\top\right]_k$, then they differ on the associated coefficient of at least one element $e \in \mathbb{W}$. Then, there is an $\hat{f}_i$ as defined above, sensitive to that element e, so that $\hat{f}_i\left(x; \left[P^\top \mathbf{m}_c^\top\right]_k, \bar{x}_k\right) = \lambda_e^c \mathbf{v} \neq \lambda_e^d \mathbf{v} = \hat{f}_i\left(x; \left[P^\top \mathbf{m}_d^\top\right]_k, \bar{x}_k\right)$. That is, $f_c(P\bar{\mathbf{x}}) \neq f_d(P\bar{\mathbf{x}})$, and we have an $f \in F_{\mathcal{G}}$ and $\bar{\mathbf{x}}$ such that $f(P\bar{\mathbf{x}}) \notin \Delta_{\mathcal{A}}^{\mathbb{Y}}$. That is, $\mathcal{A} \notin \Lambda_{\mathcal{G}}$ implies $\mathcal{A} \notin L_{F_{\mathcal{G}}}$, which completes the proof. ■

The next result is valid for systems in which the weight set allows for the existence of an annihilator. However, the monoid is almost free, in the sense that its congruence relation does not define further equivalence classes.

Theorem 5.1.11. *Consider non-trivial output vector spaces $\{\mathbb{Y}_i\}_{i \in T}$ and assume that the edges are either on a free monoid $\mathcal{M} = \langle \mathbb{W} | \rangle$ or the result of adding an annihilator to a free monoid. That is, $\mathcal{M} = \langle \{a\} \cup \mathbb{W} | E \rangle$, with*

$$E = \{w \| a = a, \quad \forall w \in \mathcal{M}\},$$

where $\mathbb{W}$ is not necessarily countable.
Consider the set of oracle components $\hat{f}_i$, $i \in T$, that are only dependent on neighbors that are in a specific state $\bar{x}_k$, of the form

$$\hat{f}_i\left(x; \sum w, \bar{x}_k\right) = \mathbf{v} \prod p(w), \quad \mathbf{v} \in \mathbb{Y}_i, \mathbf{v} \neq 0_{\mathbb{Y}_i}.$$

If a partition $\mathcal{A}$ is invariant under the subset $F_{\mathcal{G}} \subseteq \mathcal{F}_{\mathcal{G}}$ that is constructed with oracle components $\hat{f}_i$ of the type above, then $\mathcal{A}$ is balanced in $\mathcal{G}$. □

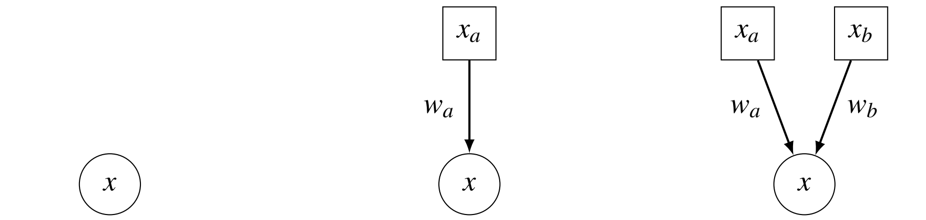
Proof. We prove this through its contrapositive. That is, $\mathcal{A} \notin \Lambda_{\mathcal{G}}$ implies $\mathcal{A} \notin L_{F_{\mathcal{G}}}$.
Consider any partition $\mathcal{A} \leq \mathcal{T}$ that is not balanced. Then, there are cells $c, d \in \mathcal{C}$ such that $\mathcal{A}(c) = \mathcal{A}(d)$ and $P^\top \mathbf{m}_c^\top \neq P^\top \mathbf{m}_d^\top$, where P is a partition matrix of $\mathcal{A}$. Consider k one of the entries in which they differ. That is, $\left[P^\top \mathbf{m}_c^\top\right]_k \neq \left[P^\top \mathbf{m}_d^\top\right]_k$.
We now choose a state in the related polydiagonal $\mathbf{x} \in \Delta_{\mathcal{A}}^{\mathbb{X}}$, that is, $\mathbf{x} = P\bar{\mathbf{x}}$ for some $\bar{\mathbf{x}}$, such that $\bar{x}_k$ is different from all other entries of $\bar{\mathbf{x}}$.
An element of the monoid $\mathcal{M}$ can be written as a finite sum over elements in $\mathbb{W}$. Call the support, that is, the elements that appear at least once in $\left[P^\top \mathbf{m}_c^\top\right]_k$ as $w_1, \ldots, w_n$ and the support of $\left[P^\top \mathbf{m}_d^\top\right]_k$ as $v_1, \ldots, v_m$. We can, with some function p, assign to each distinct element of the union of both sets, a distinct prime number, with the exception of the zero element $0_{\mathcal{M}}$ and a possible annihilator a, in which we have instead that $p(0_{\mathcal{M}}) = 1$ and $p(a) = 0$. Then, there is an $\hat{f}_i$ as defined above, such that $\hat{f}_i\left(x; \left[P^\top \mathbf{m}_c^\top\right]_k, \bar{x}_k\right) = \mathbf{v} \prod_{i=1}^n p(w_i)^{\alpha_i} \neq \mathbf{v} \prod_{j=1}^m p(v_j)^{\beta_j} = \hat{f}_i\left(x; \left[P^\top \mathbf{m}_d^\top\right]_k, \bar{x}_k\right)$. That is, $f_c(P\bar{\mathbf{x}}) \neq f_d(P\bar{\mathbf{x}})$, and we have an $f \in F_{\mathcal{G}}$ and $\bar{\mathbf{x}}$ such that $f(P\bar{\mathbf{x}}) \notin \Delta_{\mathcal{A}}^{\mathbb{Y}}$. That is, $\mathcal{A} \notin \Lambda_{\mathcal{G}}$ implies $\mathcal{A} \notin L_{F_{\mathcal{G}}}$, which completes the proof. ■

Remark 5.1.12. *Note that additional conditions on E, that defines a congruence relation of the monoid, might invalidate this approach, e.g., $w_1 \| w_2 = w_3 \| w_4$, where all weights are different.* □

5.2 Decomposition into coupling components

In this section, we develop our first decomposition scheme for oracle components in which the output sets $\{\mathbb{Y}_i\}_{i\in T}$ are vector spaces. We start by illustrating the main ideas with an example.

Example 5.2.1. *Consider cell types $T = \{1,2\}$ which denote the cell types "circle" and "square" respectively. Figure 5.1 presents a cell of type 1 with different types of inputs sets, denoted by the multi-indexes [00], [01] and [02] respectively. We assume a particular oracle component $\hat{f}_1 \in \hat{\mathcal{F}}_1$*



(a) No in-neighbors. (b) One in-neighbor. (c) Two in-neighbors.

Figure 5.1: Simple input sets.

has been chosen. Consider the input set in Figure 5.1a. This cell does not depend on anything else in the network, it evolves only according to its own internal dynamics. We define the function $f_1^{[00]} : \mathbb{X}_1 \to \mathbb{Y}_1$ as

$$f_1^{[00]}(x) := \hat{f}_1(x).$$

We use this to rewrite the function evaluation of the input set in Figure 5.1b as

$$\hat{f}_1(x; w_a, x_a) = f_1^{[00]}(x) + f_1^{[01]}(x; w_a, x_a),$$

where $f_1^{[01]} : \mathbb{X}_1 \times \mathcal{M}_{12} \times \mathbb{X}_2 \to \mathbb{Y}_1$ is defined as

$$f_1^{[01]}(x; w_a, x_a) := \hat{f}_1(x; w_a, x_a) - f_1^{[00]}(x).$$

That is, we decompose the evaluation of the oracle component $\hat{f}_1$ into the internal dynamics of the cell $\left(f_1^{[00]}\right)$ and the influence from its single in-neighbor of cell type 2 $\left(f_1^{[01]}\right)$. Note that if the weight value is 0_{12}, this case reduces to the one in Figure 5.1a, which implies that $f_1^{[01]}(x; 0_{12}, x_a) = 0_{\mathbb{Y}_1}$.

Consider now the input set in Figure 5.1c. We can write its evaluation of the oracle component as

$$\hat{f}_1\left(x; \begin{bmatrix} w_a \\ w_b \end{bmatrix}, \begin{bmatrix} x_a \\ x_b \end{bmatrix}\right) = f_1^{[00]}(x) + f_1^{[01]}(x; w_a, x_a) + f_1^{[01]}(x; w_b, x_b) + f_1^{[02]}\left(x; \begin{bmatrix} w_a \\ w_b \end{bmatrix}, \begin{bmatrix} x_a \\ x_b \end{bmatrix}\right),$$

where $f_1^{[02]} \colon \mathbb{X}_1 \times \mathcal{M}_{12}^2 \times \mathbb{X}_2^2 \to \mathbb{Y}_1$ *is defined as*

$$f_1^{[02]}\left(x; \begin{bmatrix} w_a \\ w_b \end{bmatrix}, \begin{bmatrix} x_a \\ x_b \end{bmatrix}\right) := \hat{f}_1\left(x; \begin{bmatrix} w_a \\ w_b \end{bmatrix}, \begin{bmatrix} x_a \\ x_b \end{bmatrix}\right) - f_1^{[00]}(x) - f_1^{[01]}(x; w_a, x_a) - f_1^{[01]}(x; w_b, x_b).$$

The term $f_1^{[02]}$ *describes a* 2*-order coupling effect of cells of type "square" onto cells of type "circle". By definition, it corresponds to what cannot be explained by the internal dynamics* $\left(f_1^{[00]}\right)$ *(*0*-order coupling) and the* 1*-order coupling contributions from each "square" in-neighbor* $\left(f_1^{[01]}\right)$*. Note that if* ***any*** *of its weight parameters* w_a, w_b *is* 0_{12}*, this reduces to the previous case and similarly we conclude that* $f_1^{[02]}\left(x; \begin{bmatrix} w_a \\ w_b \end{bmatrix}, \begin{bmatrix} x_a \\ x_b \end{bmatrix}\right) = 0_{\mathbb{Y}_1}$*. Moreover, note that*

$$f_1^{[02]}\left(x; \begin{bmatrix} w_a \\ w_b \end{bmatrix}, \begin{bmatrix} x_a \\ x_b \end{bmatrix}\right) = f_1^{[02]}\left(x; \begin{bmatrix} w_b \\ w_a \end{bmatrix}, \begin{bmatrix} x_b \\ x_a \end{bmatrix}\right).$$

Consider now the case where $x_a = x_b = x_{ab}$*. This is equivalent to having an edge weight of* $w_a \| w_b$ *in Figure 5.1b. This implies*

$$f_1^{[01]}(x; w_a \| w_b, x_{ab}) = f_1^{[01]}(x; w_a, x_{ab}) + f_1^{[01]}(x; w_b, x_{ab}) + f_1^{[02]}\left(x; \begin{bmatrix} w_a \\ w_b \end{bmatrix}, \begin{bmatrix} x_{ab} \\ x_{ab} \end{bmatrix}\right), \tag{5.1}$$

which means that $f_1^{[01]}$ *and* $f_1^{[02]}$ *are related to one another and cannot be chosen independently.* □

The following definition is the generalization of this approach to arbitrary finite cell types and in-neighborhoods.

Definition 5.2.2. *Consider the set of cell types* T *and the related sets* $\{\mathbb{X}_i, \mathbb{Y}_i\}_{i\in T}$ *where* $\{\mathbb{Y}_i\}_{i\in T}$ *are vector spaces. Given an oracle component* $\hat{f}_i \in \hat{\mathcal{F}}_i$*,* $i \in T$*, we define the family of* ***coupling components*** $\{f_i^{\mathbf{k}}\}_{\mathbf{k}\geq \mathbf{0}_{|T|}}$*, with*

$$f_i^{\mathbf{k}} \colon \mathbb{X}_i \times \mathcal{M}_i^{\mathbf{k}} \times \mathbb{X}^{\mathbf{k}} \to \mathbb{Y}_i, \tag{5.2}$$

defined recursively by

$$f_i^{\mathcal{K}(\mathbf{s})}(x; \mathbf{w_s}, \mathbf{x_s}) := \hat{f}_i(x; \mathbf{w_s}, \mathbf{x_s}) - \sum_{\bar{\mathbf{s}}\subset \mathbf{s}} f_i^{\mathcal{K}(\bar{\mathbf{s}})}(x; \mathbf{w}_{\bar{\mathbf{s}}}, \mathbf{x}_{\bar{\mathbf{s}}}), \tag{5.3}$$

where $\mathbf{k} = \mathcal{K}(\mathbf{s})$ *gives the corresponding multi-index of the types of cells* $\mathbf{s}$ *and* $x \in \mathbb{X}_i$*,* $\mathbf{x_s} \in \mathbb{X}^{\mathbf{k}}$*,* $\mathbf{w_s} \in \mathcal{M}_i^{\mathbf{k}}$*.* □

The following result expands the recursive formula in Equation (5.3) and writes $\{f_i^{\mathbf{k}}\}_{\mathbf{k}\geq \mathbf{0}_{|T|}}$ explicitly in terms of $\hat{f}_i$.

Lemma 5.2.3. *The coupling components $\{f_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}}$ of an oracle component $\hat{f}_i \in \hat{\mathcal{F}}_i$, $i \in T$, are given by*

$$f_i^{\mathcal{K}(\mathbf{s})}(x;\mathbf{w_s},\mathbf{x_s}) = \sum_{\bar{\mathbf{s}}\subseteq\mathbf{s}} (-1)^{|\mathbf{s}|-|\bar{\mathbf{s}}|} \hat{f}_i(x;\mathbf{w}_{\bar{\mathbf{s}}},\mathbf{x}_{\bar{\mathbf{s}}}). \tag{5.4}$$

□

Proof. The proof is by strong induction. Assume the statement to be true for all $\bar{\mathbf{s}} \subset \mathbf{s}$. Then, by assumption we can plug the explicit formula Equation (5.4) into the recursive definition Equation (5.3) in order to obtain

$$\begin{aligned} f_i^{\mathcal{K}(\mathbf{s})}(x;\mathbf{w_s},\mathbf{x_s}) &= \hat{f}_i(x;\mathbf{w_s},\mathbf{x_s}) - \sum_{\bar{\mathbf{s}}\subset\mathbf{s}} f_i^{\mathcal{K}(\bar{\mathbf{s}})}(x;\mathbf{w}_{\bar{\mathbf{s}}},\mathbf{x}_{\bar{\mathbf{s}}}) \\ &= \hat{f}_i(x;\mathbf{w_s},\mathbf{x_s}) - \sum_{\bar{\mathbf{s}}\subset\mathbf{s}}\sum_{\bar{\mathbf{r}}\subseteq\bar{\mathbf{s}}} (-1)^{|\bar{\mathbf{s}}|-|\bar{\mathbf{r}}|} \hat{f}_i(x;\mathbf{w}_{\bar{\mathbf{r}}},\mathbf{x}_{\bar{\mathbf{r}}}). \end{aligned}$$

We reorder this such that the outer sum is indexed over $\bar{\mathbf{r}}$, which yields

$$\hat{f}_i(x;\mathbf{w_s},\mathbf{x_s}) - \sum_{\bar{\mathbf{r}}\subset\mathbf{s}} \left[\sum_{\substack{\bar{\mathbf{s}}\subset\mathbf{s}\\ \bar{\mathbf{s}}\supseteq\bar{\mathbf{r}}}} (-1)^{|\bar{\mathbf{s}}|-|\bar{\mathbf{r}}|} \right] \hat{f}_i(x;\mathbf{w}_{\bar{\mathbf{r}}},\mathbf{x}_{\bar{\mathbf{r}}}).$$

Note that

$$\sum_{\substack{\bar{\mathbf{s}}\subset\mathbf{s}\\ \bar{\mathbf{s}}\supseteq\bar{\mathbf{r}}}} (-1)^{|\bar{\mathbf{s}}|-|\bar{\mathbf{r}}|} = \sum_{(\bar{\mathbf{s}}\setminus\bar{\mathbf{r}})\subset(\mathbf{s}\setminus\bar{\mathbf{r}})} (-1)^{|\bar{\mathbf{s}}\setminus\bar{\mathbf{r}}|} = \sum_{(\bar{\mathbf{s}}\setminus\bar{\mathbf{r}})\subseteq(\mathbf{s}\setminus\bar{\mathbf{r}})} (-1)^{|\bar{\mathbf{s}}\setminus\bar{\mathbf{r}}|} - (-1)^{|\mathbf{s}\setminus\bar{\mathbf{r}}|}.$$

In the power set of a non-empty finite set, half of the subsets have an even size and the other half has odd size. Therefore, if $\bar{\mathbf{r}} \subset \mathbf{s}$, we are in this situation and the sum cancels, and we get

$$\begin{aligned} f_i^{\mathcal{K}(\mathbf{s})}(x;\mathbf{w_s},\mathbf{x_s}) &= \hat{f}_i(x;\mathbf{w_s},\mathbf{x_s}) - \sum_{\bar{\mathbf{r}}\subset\mathbf{s}} \left[-(-1)^{|\mathbf{s}\setminus\bar{\mathbf{r}}|} \right] \hat{f}_i(x;\mathbf{w}_{\bar{\mathbf{r}}},\mathbf{x}_{\bar{\mathbf{r}}}) \\ &= \hat{f}_i(x;\mathbf{w_s},\mathbf{x_s}) + \sum_{\bar{\mathbf{r}}\subset\mathbf{s}} (-1)^{|\mathbf{s}|-|\bar{\mathbf{r}}|} \hat{f}_i(x;\mathbf{w}_{\bar{\mathbf{r}}},\mathbf{x}_{\bar{\mathbf{r}}}) \\ &= \sum_{\bar{\mathbf{r}}\subseteq\mathbf{s}} (-1)^{|\mathbf{s}|-|\bar{\mathbf{r}}|} \hat{f}_i(x;\mathbf{w}_{\bar{\mathbf{r}}},\mathbf{x}_{\bar{\mathbf{r}}}), \end{aligned}$$

which proves the result for $\mathbf{s}$. Note that the strong induction immediately satisfies the case $\mathbf{s} = \emptyset$ since its hypothesis is vacuously true. ■

Similarly, we can also write $\hat{f}_i$ explicitly in terms of $\{f_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}}$.

Lemma 5.2.4. *An oracle component $\hat{f}_i \in \hat{\mathcal{F}}_i$, $i \in T$ is given by its coupling components $\{f_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}}$, according to*

$$\hat{f}_i(x;\mathbf{w_s},\mathbf{x_s}) = \sum_{\bar{\mathbf{s}}\subseteq\mathbf{s}} f_i^{\mathcal{K}(\bar{\mathbf{s}})}(x;\mathbf{w}_{\bar{\mathbf{s}}},\mathbf{x}_{\bar{\mathbf{s}}}). \tag{5.5}$$

□

Proof. This is immediate from Equation (5.3) by simple rearrangement. ■

Note that Equation (5.5) can also be written as

$$\hat{f}_i(x; \mathbf{w_s}, \mathbf{x_s}) = \sum_{\substack{\bar{\mathbf{k}} \leq \mathbf{k} \\ \mathcal{K}(\mathbf{s}) = \mathbf{k}}} \sum_{\substack{\bar{\mathbf{s}} \subseteq \mathbf{s} \\ \mathcal{K}(\bar{\mathbf{s}}) = \bar{\mathbf{k}}}} f_i^{\bar{\mathbf{k}}}(x; \mathbf{w}_{\bar{\mathbf{s}}}, \mathbf{x}_{\bar{\mathbf{s}}}) . \tag{5.6}$$

Remark 5.2.5. *The number of multi-indexes smaller or equal to* $\mathbf{k}$ *is* $\prod_{i \in T}(k_i + 1)$ *and for each particular* $\bar{\mathbf{k}}$ *the number of terms in the sum is* $\prod_{i \in T} \binom{k_i}{\bar{k}_i}$. □

Remark 5.2.6. *These functions operate on an arbitrary (but finite) set of cells* $\mathbf{s}$*. Even though there is no upper bound for the amount of terms in the sums, for any particular chosen* $\mathbf{s}$ *the sum is always finite. Therefore, everything is well-defined and there are no convergence issues.* □

This is exactly the anchored decomposition Kuo et al. (2010) applied to an arbitrary finite set of variables. The decomposition is done with respect to the weights of $\mathbf{w_s}$, anchoring them at 0_{ij}, for the appropriate $j \in T$. From the properties of the anchored decomposition we know immediately that if any of the entries of $\mathbf{w}$ is 0_{ij}, then $f_i^{\mathbf{k}}(x; \mathbf{w}, \mathbf{x}) = 0_{\mathbb{Y}_i}$. From item 3 of Definition 2.4.1, we note that when we anchor some entry of $\mathbf{w_s}$ to 0_{ij} we are also removing the functional dependence on the corresponding entry of $\mathbf{x_s}$.

Moreover, note that for subsets of cells $\mathbf{s}_1, \mathbf{s}_2 \subset \mathbf{s}$ such that $\mathcal{K}(\mathbf{s}_1) = \mathcal{K}(\mathbf{s}_2) = \mathbf{k}$, we indexed their associated function by $\mathbf{k}$ instead of by $\mathbf{s}_1$ and $\mathbf{s}_2$ as is traditional in the anchored decomposition. This is proper since the functions $\{f_i^{\mathbf{k}}\}_{\mathbf{k} \geq \mathbf{0}_{|T|}}$ inherit from $\hat{f}_i$ the property of being invariant to permutations.

In summary, the decomposition according to Definition 5.2.2 gives us a family of functions $\{f_i^{\mathbf{k}}\}_{\mathbf{k} \geq \mathbf{0}_{|T|}}$, which is an equivalent representation of a given oracle component function $\hat{f}_i$.

The following result presents the necessary and sufficient conditions for $\{f_i^{\mathbf{k}}\}_{\mathbf{k} \geq \mathbf{0}_{|T|}}$ to be such that it corresponds to a valid $\hat{f}_i$. That is, for the corresponding $\hat{f}_i$ to follow Definition 2.4.1.

Theorem 5.2.7. *The family of functions* $\{f_i^{\mathbf{k}}\}_{\mathbf{k} \geq \mathbf{0}_{|T|}}$*, represents some valid oracle component* $\hat{f}_i \in \hat{\mathcal{F}}_i$*, and is related to it according to Definition 5.2.2, if and only if for every* $\mathbf{k} \geq \mathbf{0}_{|T|}$*,* $f_i^{\mathbf{k}}$ *has the following properties:*

1. *If* σ *is any permutation matrix (of appropriate dimension), then*

$$f_i^{\mathbf{k}}(x; \mathbf{w}, \mathbf{x}) = f_i^{\mathbf{k}}(x; \sigma\mathbf{w}, \sigma\mathbf{x}) . \tag{5.7}$$

2. *If $k_j > 0$, then $f_i^{\mathbf{k}}$ and $f_i^{\mathbf{k}+\mathbf{e}_j}$ are related by*

$$f_i^{\mathbf{k}}\left(x;\begin{bmatrix} w_{j_1}\|w_{j_2} \\ \mathbf{w_s}\end{bmatrix},\begin{bmatrix} x_{j_{12}} \\ \mathbf{x_s}\end{bmatrix}\right) = f_i^{\mathbf{k}}\left(x;\begin{bmatrix} w_{j_1} \\ \mathbf{w_s}\end{bmatrix},\begin{bmatrix} x_{j_{12}} \\ \mathbf{x_s}\end{bmatrix}\right) + f_i^{\mathbf{k}}\left(x;\begin{bmatrix} w_{j_2} \\ \mathbf{w_s}\end{bmatrix},\begin{bmatrix} x_{j_{12}} \\ \mathbf{x_s}\end{bmatrix}\right) + f_i^{\mathbf{k}+\mathbf{e}_j}\left(x;\begin{bmatrix} w_{j_1} \\ w_{j_2} \\ \mathbf{w_s}\end{bmatrix},\begin{bmatrix} x_{j_{12}} \\ x_{j_{12}} \\ \mathbf{x_s}\end{bmatrix}\right), \tag{5.8}$$

where $\mathbf{s}$ is a set of cells such that $\mathcal{K}(\mathbf{s}) = \mathbf{k} - \mathbf{e}_j$, and the indexes j_1, j_2 and j_{12} denote cells of type j.

3. *If the index j denotes a cell of type $j \in T$, then*

$$f_i^{\mathbf{k}}\left(x;\begin{bmatrix} 0_{ij} \\ \mathbf{w}\end{bmatrix},\begin{bmatrix} x_j \\ \mathbf{x}\end{bmatrix}\right) = 0_{\mathbb{Y}_i}. \tag{5.9}$$

□

Proof. We begin by proving the $\implies$ direction. That is, for a given $\hat{f}_i$, the family of functions $\{f_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}}$ will have the properties in items 1 to 3.

Item 1 is immediate from Equation (5.4), which writes $f_i^{\mathcal{K}(\mathbf{s})}$ explicitly as a function of $\hat{f}_i$, together with *Equation* (2.2). Note that the sum Equation (5.4) being indexed over all subsets $\bar{\mathbf{s}} \subseteq \mathbf{s}$ is crucial to keep the whole sum invariant under permutations. This means that unlike what is traditional in the general anchored decomposition, we do not require to index the coupling components according to cells subsets (e.g., $f_i^{\mathbf{s}_1}$, $f_i^{\mathbf{s}_2}$) since they are functionally the same whenever $\mathcal{K}(\mathbf{s}_1) = \mathcal{K}(\mathbf{s}_2)$. Instead we can freely index them according to their respective type multi-index. That is, our definition is self-consistent.

The proof of item 2 is by strong induction. Assume the statement to be true for all $\bar{\mathbf{s}} \subset \mathbf{s}$. We apply Equation (5.5) to Equation (2.3). Then, the left hand side becomes

$$\sum_{\bar{\mathbf{s}}\subseteq\mathbf{s}}\left(f_i^{\mathcal{K}(\bar{\mathbf{s}})}(x;\mathbf{w}_{\bar{\mathbf{s}}},\mathbf{x}_{\bar{\mathbf{s}}}) + f_i^{\mathcal{K}(\bar{\mathbf{s}})+\mathbf{e}_j}\left(x;\begin{bmatrix} w_{j_1}\|w_{j_2} \\ \mathbf{w}_{\bar{\mathbf{s}}}\end{bmatrix},\begin{bmatrix} x_{j_{12}} \\ \mathbf{x}_{\bar{\mathbf{s}}}\end{bmatrix}\right)\right),$$

and the right hand expands into

$$\sum_{\bar{\mathbf{s}}\subseteq\mathbf{s}}\left[f_i^{\mathcal{K}(\bar{\mathbf{s}})}(x;\mathbf{w}_{\bar{\mathbf{s}}},\mathbf{x}_{\bar{\mathbf{s}}}) + f_i^{\mathcal{K}(\bar{\mathbf{s}})+\mathbf{e}_j}\left(x;\begin{bmatrix} w_{j_1} \\ \mathbf{w}_{\bar{\mathbf{s}}}\end{bmatrix},\begin{bmatrix} x_{j_{12}} \\ \mathbf{x}_{\bar{\mathbf{s}}}\end{bmatrix}\right) + f_i^{\mathcal{K}(\bar{\mathbf{s}})+\mathbf{e}_j}\left(x;\begin{bmatrix} w_{j_2} \\ \mathbf{w}_{\bar{\mathbf{s}}}\end{bmatrix},\begin{bmatrix} x_{j_{12}} \\ \mathbf{x}_{\bar{\mathbf{s}}}\end{bmatrix}\right) + f_i^{\mathcal{K}(\bar{\mathbf{s}})+2\mathbf{e}_j}\left(x;\begin{bmatrix} w_{j_1} \\ w_{j_2} \\ \mathbf{w}_{\bar{\mathbf{s}}}\end{bmatrix},\begin{bmatrix} x_{j_{12}} \\ x_{j_{12}} \\ \mathbf{x}_{\bar{\mathbf{s}}}\end{bmatrix}\right)\right].$$

The first terms of the sum in both sides cancel each other. Using the assumption that item 2 holds for every index $\bar{\mathbf{s}} \subset \mathbf{s}$, the last term of the left hand side cancels with the last three terms of the right hand side. Thus, what remains is those terms indexed with $\bar{\mathbf{s}} = \mathbf{s}$, that is,

$$f_i^{\mathcal{K}(\mathbf{s})+\mathbf{e}_j}\left(x; \begin{bmatrix} w_{j_1} \| w_{j_2} \\ \mathbf{w_s} \end{bmatrix}, \begin{bmatrix} x_{j_{12}} \\ \mathbf{x_s} \end{bmatrix}\right) = f_i^{\mathcal{K}(\mathbf{s})+\mathbf{e}_j}\left(x; \begin{bmatrix} w_{j_1} \\ \mathbf{w_s} \end{bmatrix}, \begin{bmatrix} x_{j_{12}} \\ \mathbf{x_s} \end{bmatrix}\right) + f_i^{\mathcal{K}(\mathbf{s})+\mathbf{e}_j}\left(x; \begin{bmatrix} w_{j_2} \\ \mathbf{w_s} \end{bmatrix}, \begin{bmatrix} x_{j_{12}} \\ \mathbf{x_s} \end{bmatrix}\right) + f_i^{\mathcal{K}(\mathbf{s})+2\mathbf{e}_j}\left(x; \begin{bmatrix} w_{j_1} \\ w_{j_2} \\ \mathbf{w_s} \end{bmatrix}, \begin{bmatrix} x_{j_{12}} \\ x_{j_{12}} \\ \mathbf{x_s} \end{bmatrix}\right).$$

This means that item 2 applies for every $\mathbf{s}$.

Item 3 comes directly from the known properties of the anchored decomposition. We prove it explicitly for completeness sake. Split the sum in Equation (5.4) into two sums according to whenever the indexed subset contains a given cell c or not. That is,

$$f_i^{\mathcal{K}(\mathbf{s})}(x; \mathbf{w_s}, \mathbf{x_s}) = \sum_{\bar{\mathbf{s}} \subseteq \mathbf{s} \setminus \{c\}} (-1)^{|\mathbf{s}|-|\bar{\mathbf{s}}|} \hat{f}_i(x; \mathbf{w}_{\bar{\mathbf{s}}}, \mathbf{x}_{\bar{\mathbf{s}}}) + \sum_{\bar{\mathbf{s}} \subseteq \mathbf{s} \setminus \{c\}} (-1)^{|\mathbf{s}|-|\bar{\mathbf{s}} \cup \{c\}|} \hat{f}_i\left(x; \begin{bmatrix} w_c \\ \mathbf{w}_{\bar{\mathbf{s}}} \end{bmatrix}, \begin{bmatrix} x_c \\ \mathbf{x}_{\bar{\mathbf{s}}} \end{bmatrix}\right).$$

If $w_c = 0_{ij}$ for some $j \in T$, then, we can apply Equation (2.4) on the right sum, which results in

$$f_i^{\mathcal{K}(\mathbf{s})}(x; \mathbf{w_s}, \mathbf{x_s}) = \sum_{\bar{\mathbf{s}} \subseteq \mathbf{s} \setminus \{c\}} (-1)^{|\mathbf{s}|-|\bar{\mathbf{s}}|} \hat{f}_i(x; \mathbf{w}_{\bar{\mathbf{s}}}, \mathbf{x}_{\bar{\mathbf{s}}}) - \sum_{\bar{\mathbf{s}} \subseteq \mathbf{s} \setminus \{c\}} (-1)^{|\mathbf{s}|-|\bar{\mathbf{s}}|} \hat{f}_i(x; \mathbf{w}_{\bar{\mathbf{s}}}, \mathbf{x}_{\bar{\mathbf{s}}}) = 0_{\mathbb{Y}_i}.$$

We now prove the $\Longleftarrow$ direction. That is, any given family of functions $\{f_i^{\mathbf{k}}\}_{\mathbf{k} \geq \mathbf{0}_{|T|}}$ with the properties in items 1 to 3, defines a valid oracle component $\hat{f}_i$. To prove that, we show that Definition 5.2.2 will always be respected for any input.

The proof of Equation (2.2) is immediate from Equation (5.5), which writes $\hat{f}_i$ explicitly as a function of $f_i^{\mathcal{K}(\mathbf{s})}$, together with item 1. Note that the sum Equation (5.5) being indexed over all subsets $\bar{\mathbf{s}} \subseteq \mathbf{s}$ is crucial to keep the whole sum invariant under permutations.

We now prove that Equation (2.3) is satisfied. Using Equation (5.5) on its left hand side gives us

$$\sum_{\bar{\mathbf{s}} \subseteq \mathbf{s}} \left(f_i^{\mathcal{K}(\bar{\mathbf{s}})}(x; \mathbf{w}_{\bar{\mathbf{s}}}, \mathbf{x}_{\bar{\mathbf{s}}}) + f_i^{\mathcal{K}(\bar{\mathbf{s}})+\mathbf{e}_j}\left(x; \begin{bmatrix} w_{j_1} \| w_{j_2} \\ \mathbf{w}_{\bar{\mathbf{s}}} \end{bmatrix}, \begin{bmatrix} x_{j_{12}} \\ \mathbf{x}_{\bar{\mathbf{s}}} \end{bmatrix}\right)\right).$$

We now apply item 2 to the second term of the sum and we obtain

$$\sum_{\bar{\mathbf{s}} \subseteq \mathbf{s}} \Bigg[f_i^{\mathcal{K}(\bar{\mathbf{s}})}(x; \mathbf{w}_{\bar{\mathbf{s}}}, \mathbf{x}_{\bar{\mathbf{s}}}) + f_i^{\mathcal{K}(\bar{\mathbf{s}})+\mathbf{e}_j}\left(x; \begin{bmatrix} w_{j_1} \\ \mathbf{w}_{\bar{\mathbf{s}}} \end{bmatrix}, \begin{bmatrix} x_{j_{12}} \\ \mathbf{x}_{\bar{\mathbf{s}}} \end{bmatrix}\right) + f_i^{\mathcal{K}(\bar{\mathbf{s}})+\mathbf{e}_j}\left(x; \begin{bmatrix} w_{j_2} \\ \mathbf{w}_{\bar{\mathbf{s}}} \end{bmatrix}, \begin{bmatrix} x_{j_{12}} \\ \mathbf{x}_{\bar{\mathbf{s}}} \end{bmatrix}\right) + f_i^{\mathcal{K}(\bar{\mathbf{s}})+2\mathbf{e}_j}\left(x; \begin{bmatrix} w_{j_1} \\ w_{j_2} \\ \mathbf{w}_{\bar{\mathbf{s}}} \end{bmatrix}, \begin{bmatrix} x_{j_{12}} \\ x_{j_{12}} \\ \mathbf{x}_{\bar{\mathbf{s}}} \end{bmatrix}\right)\Bigg],$$

Using Equation (5.5) again gives us the right hand side of Equation (2.3).

We now prove Equation (2.4). Split the sum in Equation (5.5) into two sums according to whenever the indexed subset contains a given cell c or not. That is,

$$\hat{f}_i\left(x;\mathbf{w}_\mathbf{s},\mathbf{x}_\mathbf{s}\right)=\sum_{\bar{\mathbf{s}}\subseteq\mathbf{s}\setminus\{c\}} f_i^{\mathcal{K}(\bar{\mathbf{s}})}\left(x;\mathbf{w}_{\bar{\mathbf{s}}},\mathbf{x}_{\bar{\mathbf{s}}}\right)+\sum_{\bar{\mathbf{s}}\subseteq\mathbf{s}\setminus\{c\}} f_i^{\mathcal{K}(\bar{\mathbf{s}}\cup\{c\})}\left(x;\begin{bmatrix}w_c\\ \mathbf{w}_{\bar{\mathbf{s}}}\end{bmatrix},\begin{bmatrix}x_c\\ \mathbf{x}_{\bar{\mathbf{s}}}\end{bmatrix}\right).$$

If $w_c=0_{ij}$ for some $j\in T$, then, we can apply item 3 on the right sum, which results in

$$\hat{f}_i\left(x;\mathbf{w}_\mathbf{s},\mathbf{x}_\mathbf{s}\right)=\sum_{\bar{\mathbf{s}}\subseteq\mathbf{s}\setminus\{c\}} f_i^{\mathcal{K}(\bar{\mathbf{s}})}\left(x;\mathbf{w}_{\bar{\mathbf{s}}},\mathbf{x}_{\bar{\mathbf{s}}}\right)=\hat{f}_i\left(x;\mathbf{w}_{\mathbf{s}\setminus\{c\}},\mathbf{x}_{\mathbf{s}\setminus\{c\}}\right).$$

■

At this point, we have started with the definition of oracle components $\hat{f}_i$ in Definition 2.4.1. Then, we established a bijective correspondence between $\hat{f}_i$ and a family of coupling components $\{f_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}}$ in Definition 5.2.2. Finally, Theorem 5.2.7 completed the cycle by making it so that we can also start by first constructing a valid $\{f_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}}$ and then obtaining the corresponding $\hat{f}_i$ afterwards.

We are now interested in knowing how to manipulate this mathematical object through this new representation. Subsequently, we will provide some examples that illustrate this decomposition and its properties.

Lemma 5.2.8. *Consider the oracle components $\hat{f}_i$ and the ones in the sequence $({}^N\hat{f}_i)_{N\in\mathbb{N}}$ such that their corresponding coupling components are, respectively, $\{f_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}}$ and $\left(\{{}^Nf_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}}\right)_{N\in\mathbb{N}}$. If the output set $\mathbb{Y}_i$ is a Hausdorff topological vector space, then,*

$$\lim_{N\to\infty}{}^N\hat{f}_i=\hat{f}_i\iff\lim_{N\to\infty}\{{}^Nf_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}}=\{f_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}}$$

in the topology of pointwise convergence. □

Proof. We begin by proving the $\Longrightarrow$ direction. That is, assume $\lim_{N\to\infty}{}^N\hat{f}_i=\hat{f}_i$.
For any $N\in\mathbb{N}$, we know from Equation (5.4), that for any set of cells $\mathbf{s}$

$$\begin{aligned}\lim_{N\to\infty}{}^Nf_i^{\mathcal{K}(\mathbf{s})}\left(x;\mathbf{w}_\mathbf{s},\mathbf{x}_\mathbf{s}\right)&=\lim_{N\to\infty}\sum_{\bar{\mathbf{s}}\subseteq\mathbf{s}}(-1)^{|\mathbf{s}|-|\bar{\mathbf{s}}|N}\hat{f}_i\left(x;\mathbf{w}_{\bar{\mathbf{s}}},\mathbf{x}_{\bar{\mathbf{s}}}\right)\\&=\sum_{\bar{\mathbf{s}}\subseteq\mathbf{s}}(-1)^{|\mathbf{s}|-|\bar{\mathbf{s}}|}\lim_{N\to\infty}{}^N\hat{f}_i\left(x;\mathbf{w}_{\bar{\mathbf{s}}},\mathbf{x}_{\bar{\mathbf{s}}}\right)\\&=\sum_{\bar{\mathbf{s}}\subseteq\mathbf{s}}(-1)^{|\mathbf{s}|-|\bar{\mathbf{s}}|}\hat{f}_i\left(x;\mathbf{w}_{\bar{\mathbf{s}}},\mathbf{x}_{\bar{\mathbf{s}}}\right)\\&=f_i^{\mathcal{K}(\mathbf{s})}\left(x;\mathbf{w}_\mathbf{s},\mathbf{x}_\mathbf{s}\right).\end{aligned}$$

We now prove the $\Longleftarrow$ direction. That is, assume $\lim_{N\to\infty}\{{}^N f_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}} = \{f_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}}$.
For any $N\in\mathbb{N}$, we know from Equation (5.5), that for any set of cells $\mathbf{s}$

$$\begin{aligned}\lim_{N\to\infty} {}^N\hat{f}_i\left(x;\mathbf{w}_\mathbf{s},\mathbf{x}_\mathbf{s}\right) &= \lim_{N\to\infty}\sum_{\bar{\mathbf{s}}\subseteq\mathbf{s}} {}^N f_i^{\mathcal{K}(\bar{\mathbf{s}})}\left(x;\mathbf{w}_{\bar{\mathbf{s}}},\mathbf{x}_{\bar{\mathbf{s}}}\right)\\ &= \sum_{\bar{\mathbf{s}}\subseteq\mathbf{s}}\lim_{N\to\infty} {}^N f_i^{\mathcal{K}(\bar{\mathbf{s}})}\left(x;\mathbf{w}_{\bar{\mathbf{s}}},\mathbf{x}_{\bar{\mathbf{s}}}\right)\\ &= \sum_{\bar{\mathbf{s}}\subseteq\mathbf{s}} f_i^{\mathcal{K}(\bar{\mathbf{s}})}\left(x;\mathbf{w}_{\bar{\mathbf{s}}},\mathbf{x}_{\bar{\mathbf{s}}}\right)\\ &= \hat{f}_i\left(x;\mathbf{w}_\mathbf{s},\mathbf{x}_\mathbf{s}\right).\end{aligned}$$

Note that in a topological vector space the addition operation $+(\cdot,\cdot)$ is (jointly) continuous. This is what allowed us to convert limits of (finite) sums into (finite) sums of the limits. The Hausdorff property is required to ensure that the limits are always as stated due to uniqueness. ∎

Lemma 5.2.9. *For two oracle components $\hat{f}_i,\hat{g}_i\in\hat{\mathcal{F}}_i$ with coupling components $\{f_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}}$ and $\{g_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}}$ respectively, the coupling components of $\hat{h}_i=\alpha\hat{f}_i+\hat{g}_i$ are given by $\{\alpha f_i^{\mathbf{k}}+g_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}}$, for any scalar α.* □

Proof. This comes directly from writing the coupling components explicitly in terms of the oracle components as in Equation (5.4). That is,

$$\begin{aligned}h_i^{\mathcal{K}(\mathbf{s})}\left(x;\mathbf{w}_\mathbf{s},\mathbf{x}_\mathbf{s}\right) &= \sum_{\bar{\mathbf{s}}\subseteq\mathbf{s}}(-1)^{|\mathbf{s}|-|\bar{\mathbf{s}}|}\left(\alpha\hat{f}_i+\hat{g}_i\right)\left(x;\mathbf{w}_{\bar{\mathbf{s}}},\mathbf{x}_{\bar{\mathbf{s}}}\right)\\ &= \alpha\left(\sum_{\bar{\mathbf{s}}\subseteq\mathbf{s}}(-1)^{|\mathbf{s}|-|\bar{\mathbf{s}}|}\hat{f}_i\left(x;\mathbf{w}_{\bar{\mathbf{s}}},\mathbf{x}_{\bar{\mathbf{s}}}\right)\right)+\sum_{\bar{\mathbf{s}}\subseteq\mathbf{s}}(-1)^{|\mathbf{s}|-|\bar{\mathbf{s}}|}\hat{g}_i\left(x;\mathbf{w}_{\bar{\mathbf{s}}},\mathbf{x}_{\bar{\mathbf{s}}}\right)\\ &= \alpha f_i^{\mathcal{K}(\mathbf{s})}\left(x;\mathbf{w}_\mathbf{s},\mathbf{x}_\mathbf{s}\right)+g_i^{\mathcal{K}(\mathbf{s})}\left(x;\mathbf{w}_\mathbf{s},\mathbf{x}_\mathbf{s}\right)\\ &= \left(\alpha f_i^{\mathcal{K}(\mathbf{s})}+g_i^{\mathcal{K}(\mathbf{s})}\right)\left(x;\mathbf{w}_\mathbf{s},\mathbf{x}_\mathbf{s}\right).\end{aligned}$$

∎

We have shown that operating linearly on $\hat{\mathcal{F}}_i$ is completely straightforward, with the coupling components $\{f_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}}$ being affected component-wise according to the respective linear combination.

Corollary 5.2.10. *The coupling components of order 0, that is, $f_i^{\mathbf{0}}$, which describes the inner dynamics of a cell, are completely free and independent of the remaining coupling components $\{f_i^{\mathbf{k}}\}_{\mathbf{k}>\mathbf{0}}$.* □

Corollary 5.2.11. *Consider $f_i^{\mathbf{k}+\mathbf{e}_j}=0_{\mathbb{Y}_i}$ for some $\mathbf{k}\geq\mathbf{0}_{|T|}$, with $k_j\geq 1$, $j\in T$.*
*Then, $f_i^{\mathbf{k}}$ is **additive in the weights** with respect to type j. That is,*

$$f_i^{\mathbf{k}}\left(x;\begin{bmatrix}w_{j_1}\|w_{j_2}\\ \mathbf{w}_\mathbf{s}\end{bmatrix},\begin{bmatrix}x_{j_{12}}\\ \mathbf{x}_\mathbf{s}\end{bmatrix}\right)=f_i^{\mathbf{k}}\left(x;\begin{bmatrix}w_{j_1}\\ \mathbf{w}_\mathbf{s}\end{bmatrix},\begin{bmatrix}x_{j_{12}}\\ \mathbf{x}_\mathbf{s}\end{bmatrix}\right)+f_i^{\mathbf{k}}\left(x;\begin{bmatrix}w_{j_2}\\ \mathbf{w}_\mathbf{s}\end{bmatrix},\begin{bmatrix}x_{j_{12}}\\ \mathbf{x}_\mathbf{s}\end{bmatrix}\right). \tag{5.10}$$

□

The coupling decomposition allows us to define very important concepts that will prove essential in Section 5.3.

Definition 5.2.12. *We say that an oracle component $\hat{f}_i \in \hat{\mathcal{F}}_i$ with coupling components $\{f_i^{\mathbf{k}}\}_{\mathbf{k}\geq \mathbf{0}_{|T|}}$ has (finite)* ***coupling order*** *$\gamma_j \in \mathbb{N}_0$ with respect to the cell type $j \in T$, if there is some $\mathbf{k} \geq \mathbf{0}_{|T|}$, with $k_j = \gamma_j$ such that $f_i^{\mathbf{k}} \neq 0_{\mathbb{Y}_i}$ and there is no such $\overline{\mathbf{k}} \geq \mathbf{0}_{|T|}$ with $\overline{k_j} > \gamma_j$.*
We say that $\hat{f}_i \in \hat{\mathcal{F}}_i$ has infinite coupling order ($\gamma_j = \infty$) with respect to the cell type $j \in T$, if for every $k_j \in \mathbb{N}_0$ there is some $\overline{\mathbf{k}} \geq \mathbf{0}_{|T|}$ such that $f_i^{\overline{\mathbf{k}}} \neq 0_{\mathbb{Y}_i}$, with $\overline{k_j} \geq k_j$.
In particular, if $\gamma_j = 1$ or $\gamma_j = 0$, we say that it is ***additive*** *or* ***uncoupled****, respectively, with regard to $j \in T$.* □

Corollary 5.2.13. *Consider an oracle component $\hat{f}_i \in \hat{\mathcal{F}}_i$ with finite coupling order $\gamma_j \geq 1$ for some $j \in T$. Then, for any $\mathbf{k} \geq \mathbf{0}_{|T|}$ such that $k_j = \gamma_j$, $f_i^{\mathbf{k}}$ is additive in the weights with respect to type j.* □

Proof. If is it of order $k_j = \gamma_j$, then, $f_i^{\mathbf{k}+\mathbf{e}_j} = 0_{\mathbb{Y}_i}$. The rest follows from Corollary 5.2.11. ■

Lemma 5.2.14. *Consider an oracle component $\hat{f}_i \in \hat{\mathcal{F}}_i$ such that for a particular $j \in T$ the associated commutative monoid $\mathcal{M}_{ij}$ has an annihilator a_{ij}. Then the coupling order of $\hat{f}_i$ with respect to cell type $j \in T$, is either infinite or 0 (uncoupled).* □

Proof. The proof is by contradiction. Assume $\hat{f}_i$ has finite order $\gamma_j \geq 1$. Then, for every $\mathbf{k} \geq \mathbf{0}_{|T|}$, with $k_j = \gamma_j$, we have that $f_i^{\mathbf{k}+\mathbf{e}_j} = 0_{\mathbb{Y}_i}$. From Corollary 5.2.13, $f_i^{\mathbf{k}}$ is additive, which implies

$$f_i^{\mathbf{k}}\left(x;\begin{bmatrix} w_{j_1}\|a_{ij} \\ \mathbf{w_s}\end{bmatrix},\begin{bmatrix} x_{j_{12}} \\ \mathbf{x_s}\end{bmatrix}\right) = f_i^{\mathbf{k}}\left(x;\begin{bmatrix} w_{j_1} \\ \mathbf{w_s}\end{bmatrix},\begin{bmatrix} x_{j_{12}} \\ \mathbf{x_s}\end{bmatrix}\right) + f_i^{\mathbf{k}}\left(x;\begin{bmatrix} a_{ij} \\ \mathbf{w_s}\end{bmatrix},\begin{bmatrix} x_{j_{12}} \\ \mathbf{x_s}\end{bmatrix}\right). \tag{5.11}$$

Since $w_{j_1}\|a_{ij} = a_{ij}$, this means that $f_i^{\mathbf{k}} = 0_{\mathbb{Y}_i}$, which contradicts the assumption that $\hat{f}_i$ is of order γ_j. ■

We illustrate the decomposition into coupling components scheme with the following examples.

Example 5.2.15. *Consider a single-type network such that*

$$\hat{f}_i(x;\mathbf{w_s},\mathbf{x_s}) = f_i^0(x) + \left(\sum_{c\in\mathbf{s}} w_c x_c\right)^2.$$

We can derive the commutative monoid that defines the edge merging. It has to obey

$$f_i^0(x) + ((w_1\|w_2)x_{12})^2 = f_i^0(x) + (w_1x_{12}+w_2x_{12})^2,$$
$$(w_1\|w_2)^2 x_{12}^2 = (w_1+w_2)^2 x_{12}^2,$$

from which we conclude that $w_1 \| w_2$ *is either* $w_1 + w_2$ *or* $-(w_1 + w_2)$*. Note that for either case* $0 \| 0 = 0$*. Assume the second option to be true. From the properties of the commutative monoid*

$$\begin{aligned} w \| (0 \| 0) &= (w \| 0) \| 0, \\ w \| 0 &= -w \| 0, \\ -w &= w. \end{aligned}$$

That is, the second option will only allow the trivial situation in which all edges are 0*. Therefore, we choose* $w_1 \| w_2 = w_1 + w_2$*. Considering only one in-neighbor, we conclude that*

$$f_i^1(x; w_1, x_1) = (w_1 x_1)^2.$$

Similarly,

$$f_i^2\left(x; \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}, \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = 2(w_1 x_1)(w_2 x_2).$$

We can verify that item 2 of Theorem 5.2.7 is satisfied, that is

$$\begin{aligned} & f_i^1(x; w_1 \| w_2, x_{12}) = f_i^1(x; w_1, x_{12}) + f_i^1(x; w_2, x_{12}) + f_i^2\left(x; \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}, \begin{bmatrix} x_{12} \\ x_{12} \end{bmatrix}\right), \\ & ((w_1 + w_2) x_{12})^2 = (w_1 x_{12})^2 + (w_2 x_{12})^2 + 2 w_1 w_2 x_{12}^2, \end{aligned}$$

which is indeed true. It can be seen that higher orders will all be 0*. That is,*

$$\begin{aligned} \hat{f}_i(x; \mathbf{w_s}, \mathbf{x_s}) &= f_i^0(x) + \sum_{c \in \mathbf{s}} (w_c x_c)^2 + \sum_{\substack{c,d \in \mathbf{s} \\ c \neq d}} 2(w_c x_c)(w_d x_d) \\ &= f_i^0(x) + \sum_{c \in \mathbf{s}} f_i^1(x; w_c, x_c) + \sum_{\substack{c,d \in \mathbf{s} \\ c \neq d}} f_i^2\left(x; \begin{bmatrix} w_c \\ w_d \end{bmatrix}, \begin{bmatrix} x_c \\ x_d \end{bmatrix}\right). \end{aligned}$$

□

We now extend the previous example to a general integer power. This requires the following generalization of the binomial coefficient.

Definition 5.2.16. *Consider* $n \geq 0$ *and* $\mathbf{m} \in \mathbb{Z}^k$ *such that* $k > 1$ *and* $|\mathbf{m}| = n$*. The* ***multinomial coefficient*** $\binom{n}{\mathbf{m}}$ *is defined as*

$$\binom{n}{\mathbf{m}} := \begin{cases} \frac{n!}{\prod_{i=1}^k m_i!} & \text{if } \mathbf{m} \geq \mathbf{0}_k, \\ 0 & \text{otherwise}. \end{cases} \tag{5.12}$$

□

Remark 5.2.17. *The reason for considering the cases* $\mathbf{m} \in \mathbb{Z}^k$ *that are outside* $\mathbb{N}_0^k$ *and defining them as* 0 *is because it greatly simplifies the use of the recurrence relation*

$$\binom{n}{\mathbf{m}} = \sum_{i=1}^{k} \binom{n-1}{\mathbf{m}-1_i}, \quad n > 0. \tag{5.13}$$

This avoids having to treat many corner cases as special. For instance, in the binomial case, defined as $\binom{n}{m} = \binom{n}{m,n-m}$, *this corresponds to* $\binom{n}{m} = \binom{n-1}{m-1} + \binom{n-1}{m}$, *for* $n > 0$. *The cases* $m = 0$ *and* $m = n$ *give us* $\binom{n}{0} = \binom{n-1}{-1} + \binom{n-1}{0} = \binom{n-1}{0}$ *and* $\binom{n}{n} = \binom{n-1}{n-1} + \binom{n-1}{n} = \binom{n-1}{n-1}$, *respectively.* □

Example 5.2.18. *Consider a single-type network such that*

$$\hat{f}_i\left(x; \mathbf{w_s}, \mathbf{x_s}\right) = f_i^0(x) + \left(\sum_{c \in \mathbf{s}} w_c x_c\right)^n,$$

with $n \in \mathbb{N}$. *Then, the coupling components* f^k *for* $k > 0$ *are given according to*

$$f_i^{|\mathbf{s}|}\left(x; \mathbf{w_s}, \mathbf{x_s}\right) = \sum_{\substack{\mathbf{m} \geq \mathbf{1}_{|\mathbf{s}|} \\ |\mathbf{m}|=n}} \binom{n}{\mathbf{m}} \prod_{c \in \mathbf{s}} (w_c x_c)^{m_c} = n! \sum_{\substack{\mathbf{m} \geq \mathbf{1}_{|\mathbf{s}|} \\ |\mathbf{m}|=n}} \prod_{c \in \mathbf{s}} \frac{(w_c x_c)^{m_c}}{m_c!}.$$

The proof is by strong induction. Assume this to be true for $k \in \{1, \ldots, a-1\}$, *with* $a > 0$. *Choose any set of cells* $\mathbf{s}$ *such that* $|\mathbf{s}| = a$. *From the recursive definition we have*

$$\begin{aligned}
f_i^a\left(x; \mathbf{w_s}, \mathbf{x_s}\right) &= \hat{f}_i\left(x; \mathbf{w_s}, \mathbf{x_s}\right) - \sum_{\bar{\mathbf{s}} \subset \mathbf{s}} f_i^{|\bar{\mathbf{s}}|}\left(x; \mathbf{w}_{\bar{\mathbf{s}}}, \mathbf{x}_{\bar{\mathbf{s}}}\right) \\
&= f_i^0(x) + \sum_{\substack{\mathbf{m} \geq \mathbf{0}_{|\mathbf{s}|} \\ |\mathbf{m}|=n}} \binom{n}{\mathbf{m}} \prod_{c \in \mathbf{s}} (w_c x_c)^{m_c} - \left[f_i^0(x) + \sum_{\substack{\bar{\mathbf{s}} \subset \mathbf{s} \\ \bar{\mathbf{s}} \neq \emptyset}} \sum_{\substack{\mathbf{m} \geq \mathbf{1}_{|\bar{\mathbf{s}}|} \\ |\mathbf{m}|=n}} \binom{n}{\mathbf{m}} \prod_{c \in \bar{\mathbf{s}}} (w_c x_c)^{m_c} \right] \\
&= \sum_{\substack{\mathbf{m} \geq \mathbf{1}_{|\mathbf{s}|} \\ |\mathbf{m}|=n}} \binom{n}{\mathbf{m}} \prod_{c \in \mathbf{s}} (w_c x_c)^{m_c} \\
&= n! \sum_{\substack{\mathbf{m} \geq \mathbf{1}_{|\mathbf{s}|} \\ |\mathbf{m}|=n}} \prod_{c \in \mathbf{s}} \frac{(w_c x_c)^{m_c}}{m_c!}.
\end{aligned}$$

That is, the case $k = a$ *is also satisfied, which concludes the proof. Note that the case* $k = 1$ *comes for free due to using strong induction (its hypothesis is vacuously true), although it is trivial to verify.* □

Remark 5.2.19. *Note that for* $k > n$ *there are no multi-indexes that satisfy simultaneously* $\mathbf{m} \geq \mathbf{1}_k$ *and* $|\mathbf{m}| = n$. *Therefore,* $f_i^k = 0$ *for such* k. *The coupling order is then* $\gamma = n$. □

Remark 5.2.20. *As a sanity check we verify that item 2 of Theorem 5.2.7 is satisfied. First we can derive the commutative monoid that defines the edge merging. It has to obey*

$$f_i^0(x) + ((w_1 \| w_2) x_{12})^n = f_i^0(x) + (w_1 x_{12} + w_2 x_{12})^n,$$
$$(w_1 \| w_2)^n x_{12}^n = (w_1 + w_2)^n x_{12}^n.$$

If n is odd, then $w_1 \| w_2 = w_1 + w_2$. If n is even, we are in the same situation as in Example 5.2.15 and $w_1 \| w_2 = w_1 + w_2$ for us to be in a non-trivial setting. Now, to verify

$$f_i^{|\bar{\mathbf{s}}|+2}\left(x; \begin{bmatrix} w_1 \\ w_2 \\ \mathbf{w}_{\bar{\mathbf{s}}} \end{bmatrix}, \begin{bmatrix} x_{12} \\ x_{12} \\ \mathbf{x}_{\bar{\mathbf{s}}} \end{bmatrix}\right) = f_i^{|\bar{\mathbf{s}}|+1}\left(x; \begin{bmatrix} w_1 \| w_2 \\ \mathbf{w}_{\bar{\mathbf{s}}} \end{bmatrix}, \begin{bmatrix} x_{12} \\ \mathbf{x}_{\bar{\mathbf{s}}} \end{bmatrix}\right)$$
$$- f_i^{|\bar{\mathbf{s}}|+1}\left(x; \begin{bmatrix} w_1 \\ \mathbf{w}_{\bar{\mathbf{s}}} \end{bmatrix}, \begin{bmatrix} x_{12} \\ \mathbf{x}_{\bar{\mathbf{s}}} \end{bmatrix}\right) - f_i^{|\bar{\mathbf{s}}|+1}\left(x; \begin{bmatrix} w_2 \\ \mathbf{w}_{\bar{\mathbf{s}}} \end{bmatrix}, \begin{bmatrix} x_{12} \\ \mathbf{x}_{\bar{\mathbf{s}}} \end{bmatrix}\right),$$

we note that

$$\binom{n}{m_1, m_2, \overline{\mathbf{m}}} = \binom{m_1 + m_2}{m_1, m_2} \binom{n}{m_1 + m_2, \overline{\mathbf{m}}}.$$

Using this, the left hand side can be written as

$$\sum_{m_1, m_2 \geq 1} \binom{m_1 + m_2}{m_1, m_2} w_1^{m_1} w_2^{m_2} x_{12}^{m_1 + m_2} \sum_{\substack{\overline{\mathbf{m}} \geq \mathbf{1}_{|\bar{\mathbf{s}}|} \\ m_1 + m_2 + |\overline{\mathbf{m}}| = n}} \binom{n}{m_1 + m_2, \overline{\mathbf{m}}} \prod_{c \in \bar{\mathbf{s}}} (w_c x_c)^{m_c}$$

and the right hand side as

$$\sum_{m_{12} \geq 1} ((w_1 + w_2)^{m_{12}} - w_1^{m_{12}} - w_2^{m_{12}}) x_{12}^{m_{12}} \sum_{\substack{\overline{\mathbf{m}} \geq \mathbf{1}_{|\bar{\mathbf{s}}|} \\ m_{12} + |\overline{\mathbf{m}}| = n}} \binom{n}{m_{12}, \overline{\mathbf{m}}} \prod_{c \in \bar{\mathbf{s}}} (w_c x_c)^{m_c}.$$

Using the binomial theorem on $(w_1 + w_2)^{m_{12}}$ we see that for a fixed m_{12} we have that

$$(w_1 + w_2)^{m_{12}} - w_1^{m_{12}} - w_2^{m_{12}} = \sum_{\substack{m_1, m_2 \geq 1 \\ m_1 + m_2 = m_{12}}} \binom{m_{12}}{m_1, m_2} w_1^{m_1} w_2^{m_2}.$$

Therefore, both sides are the same. □

We now extend Example 5.2.18 to the polynomial case.

Example 5.2.21. *From Example 5.2.18 and Lemma 5.2.9 we have that for single-type networks such that*

$$\hat{f}_i(x; \mathbf{w_s}, \mathbf{x_s}) = f_i^0(x) + \sum_{n=1}^{N} a_n \left(\sum_{c \in \mathbf{s}} w_c x_c \right)^n,$$

the coupling components f_i^k for $k > 0$ are given according to

$$f_i^{|\mathbf{s}|}(x; \mathbf{w_s}, \mathbf{x_s}) = \sum_{n=1}^{N} a_n n! \sum_{\substack{\mathbf{m} \geq \mathbf{1}_{|\mathbf{s}|} \\ |\mathbf{m}|=n}} \prod_{c \in \mathbf{s}} \frac{(w_c x_c)^{m_c}}{m_c!}.$$

□

Example 5.2.22. *Consider the exponential case*

$$\hat{f}_i(x; \mathbf{w_s}, \mathbf{x_s}) = f_i^0(x) + \exp\left(\sum_{c \in \mathbf{s}} w_c x_c\right) - 1.$$

The coupling components f^k for $k > 0$ are given according to

$$f_i^{|\mathbf{s}|}(x; \mathbf{w_s}, \mathbf{x_s}) = \prod_{c \in \mathbf{s}} \left(\exp(w_c x_c) - 1\right).$$

This is proven by creating the sequence of oracle components $({}^N\hat{f}_i)_{N \in \mathbb{N}}$ such that

$${}^N\hat{f}_i(x; \mathbf{w_s}, \mathbf{x_s}) = f_i^0(x) + \sum_{n=1}^{N} \frac{1}{n!} \left(\sum_{c \in \mathbf{s}} w_c x_c\right)^n.$$

That is, the oracle components obtained by replacing $\exp(\cdot) - 1$ by its N^{th} order Taylor series truncation. From Example 5.2.21 we know that for the sequence $\left(\{{}^N f_i^k\}_{k \geq 0}\right)_{N \in \mathbb{N}}$, the components ${}^N f^k$, for $k > 0$ are given according to

$${}^N f_i^{|\mathbf{s}|}(x; \mathbf{w_s}, \mathbf{x_s}) = \sum_{\substack{\mathbf{m} \geq \mathbf{1}_{|\mathbf{s}|} \\ |\mathbf{m}| \leq N}} \prod_{c \in \mathbf{s}} \frac{(w_c x_c)^{m_c}}{m_c!}.$$

Since we know that $\lim_{N \to \infty} {}^N\hat{f}_i = \hat{f}_i$ (pointwise), from Lemma 5.2.8 we conclude that $\lim_{N \to \infty} {}^N f_i^{|\mathbf{s}|} = f_i^{|\mathbf{s}|}$. That is,

$$f_i^{|\mathbf{s}|}(x; \mathbf{w_s}, \mathbf{x_s}) = \sum_{\mathbf{m} \geq \mathbf{1}_{|\mathbf{s}|}} \prod_{c \in \mathbf{s}} \frac{(w_c x_c)^{m_c}}{m_c!}.$$

Here, the infinite sum $\sum_{\mathbf{m} \geq \mathbf{1}_{|\mathbf{s}|}}$ is taken as

$$\sum_{\mathbf{m} \geq \mathbf{1}_{|\mathbf{s}|}} := \lim_{N \to \infty} \sum_{\substack{\mathbf{m} \geq \mathbf{1}_{|\mathbf{s}|} \\ |\mathbf{m}| \leq N}}.$$

We can prove, however, that this particular infinite sum is absolutely convergent on the index set

$\mathbf{m} \geq \mathbf{1}_{|\mathbf{s}|}$. *That is,*

$$\sum_{\mathbf{m}\geq\mathbf{1}_{|\mathbf{s}|}} \left|\prod_{c\in\mathbf{s}} \frac{(w_c x_c)^{m_c}}{m_c!}\right| = \sum_{\mathbf{m}\geq\mathbf{1}_{|\mathbf{s}|}} \prod_{c\in\mathbf{s}} \frac{|w_c x_c|^{m_c}}{m_c!} = \prod_{c\in\mathbf{s}} \sum_{m_c\geq 1} \frac{|w_c x_c|^{m_c}}{m_c!} = \prod_{c\in\mathbf{s}} (\exp(|w_c x_c|) - 1) < \infty.$$

This means that the order does not matter and we can freely rearrange the sum into

$$f_i^{|\mathbf{s}|}(x;\mathbf{w_s},\mathbf{x_s}) = \sum_{\mathbf{m}\geq\mathbf{1}_{|\mathbf{s}|}} \prod_{c\in\mathbf{s}} \frac{(w_c x_c)^{m_c}}{m_c!} = \prod_{c\in\mathbf{s}} \sum_{m_c\geq 1} \frac{(w_c x_c)^{m_c}}{m_c!} = \prod_{c\in\mathbf{s}} (\exp(w_c x_c) - 1).$$

□

We now extend the previous results for multi-type networks.

Example 5.2.23. *Consider a multi-type network such that*

$$\hat{f}_i(x;\mathbf{w_s},\mathbf{x_s}) = f_i^{\mathbf{0}}(x) + \prod_{j\in T}\left(\sum_{c\in\mathbf{s}_j} w_c x_c\right)^{n_j}.$$

with $\mathbf{n} > \mathbf{0}_{|T|}$, and where $\mathbf{s}_j \subseteq \mathbf{s}$ represents the subset of cells that are of type $j \in T$. We use the definition $0^0 = 1$, which is standard and avoids many corner cases (e.g., consider the binomial theorem applied to $(x+0)^n$). Then, the coupling components $\{f^{\mathbf{k}}\}_{\mathbf{k}>\mathbf{0}_{|T|}}$ are given according to

$$f_i^{\mathcal{K}(\mathbf{s})}(x;\mathbf{w_s},\mathbf{x_s}) = \prod_{j\in T} n_j! \sum_{\substack{\mathbf{m}\geq\mathbf{1}_{|\mathbf{s}_j|}\\ |\mathbf{m}|=n_j}} \prod_{c\in\mathbf{s}_j} \frac{(w_c x_c)^{m_c}}{m_c!}.$$

Note that expanding the outer product gives us

$$\sum_{\substack{\mathbf{m}\geq\mathbf{1}_{|\mathbf{s}|}\\ |\mathbf{m}_1|=n_1\\ \vdots\\ |\mathbf{m}_{|T|}|=n_{|T|}}} \left(\prod_{j\in T} n_j!\right) \prod_{c\in\mathbf{s}} \frac{(w_c x_c)^{m_c}}{m_c!}, \quad \text{with } \mathbf{m} := \begin{bmatrix} \mathbf{m}_1 \\ \vdots \\ \mathbf{m}_{|T|} \end{bmatrix}.$$

This is now proven by strong induction. Assume this to be true for $|\mathbf{k}| \in \{1,\ldots,a-1\}$, with $a > 0$. For any $\mathbf{k} > \mathbf{0}_{|T|}$ with $|\mathbf{k}| = a$, choose any set of cells $\mathbf{s} := \{\mathbf{s}_1 \cup \ldots \cup \mathbf{s}_{|T|}\}$, such that for every $j \in T$, $\mathbf{s}_j$ is a set of cells of type j and $|\mathbf{s}_j| = k_j$.

From the recursive definition, $f_i^{\mathcal{K}(\mathbf{s})}(x;\mathbf{w_s},\mathbf{x_s}) = \hat{f}_i(x;\mathbf{w_s},\mathbf{x_s}) - \sum_{\bar{\mathbf{s}}\subset\mathbf{s}} f_i^{\mathcal{K}(\bar{\mathbf{s}})}(x;\mathbf{w}_{\bar{\mathbf{s}}},\mathbf{x}_{\bar{\mathbf{s}}})$ *becomes*

$$\begin{aligned}
&\prod_{j\in T}\sum_{\substack{\mathbf{m}\geq\mathbf{0}_{|\mathbf{s}_j|}\\|\mathbf{m}|=n_j}}\binom{n_j}{\mathbf{m}_j}\prod_{c\in\mathbf{s}_j}(w_cx_c)^{m_c} - \left[\sum_{\substack{\bar{\mathbf{s}}\subset\mathbf{s}\\\bar{\mathbf{s}}\neq\emptyset}}\sum_{\substack{\mathbf{m}\geq\mathbf{1}_{|\bar{\mathbf{s}}|}\\|\mathbf{m}_1|=n_1\\\vdots\\|\mathbf{m}_{|T|}|=n_{|T|}}}\left(\prod_{j\in T}n_j!\right)\prod_{c\in\bar{\mathbf{s}}}\frac{(w_cx_c)^{m_c}}{m_c!}\right]\\
&= \sum_{\substack{\mathbf{m}\geq\mathbf{0}_{|\mathbf{s}|}\\|\mathbf{m}_1|=n_1\\\vdots\\|\mathbf{m}_{|T|}|=n_{|T|}}}\left(\prod_{j\in T}n_j!\right)\prod_{c\in\mathbf{s}}\frac{(w_cx_c)^{m_c}}{m_c!} - \sum_{\substack{\bar{\mathbf{s}}\subset\mathbf{s}\\\bar{\mathbf{s}}\neq\emptyset}}\sum_{\substack{\mathbf{m}\geq\mathbf{1}_{|\bar{\mathbf{s}}|}\\|\mathbf{m}_1|=n_1\\\vdots\\|\mathbf{m}_{|T|}|=n_{|T|}}}\left(\prod_{j\in T}n_j!\right)\prod_{c\in\bar{\mathbf{s}}}\frac{(w_cx_c)^{m_c}}{m_c!}\\
&= \sum_{\substack{\mathbf{m}\geq\mathbf{1}_{|\mathbf{s}|}\\|\mathbf{m}_1|=n_1\\\vdots\\|\mathbf{m}_{|T|}|=n_{|T|}}}\left(\prod_{j\in T}n_j!\right)\prod_{c\in\mathbf{s}}\frac{(w_cx_c)^{m_c}}{m_c!}\\
&= \prod_{j\in T}n_j!\sum_{\substack{\mathbf{m}\geq\mathbf{1}_{|\mathbf{s}_j|}\\|\mathbf{m}|=n_j}}\prod_{c\in\mathbf{s}_j}\frac{(w_cx_c)^{m_c}}{m_c!}.
\end{aligned}$$

That is, the case $|\mathbf{k}| = a$ *is also satisfied, which concludes the proof.* □

Example 5.2.24. *From Example 5.2.23 and Lemma 5.2.9, we have that for multi-type networks such that*

$$\hat{f}_i(x;\mathbf{w_s},\mathbf{x_s}) = f_i^{\mathbf{0}}(x) + \sum_{\mathbf{n}>\mathbf{0}_{|T|}} a_{\mathbf{n}}\prod_{j\in T}\left(\sum_{c\in\mathbf{s}_j}w_cx_c\right)^{n_j},$$

with $\{a_{\mathbf{n}}\}_{\mathbf{n}>\mathbf{0}_{|T|}}$ *with finite support, the coupling components* $\{f^{\mathbf{k}}\}_{\mathbf{k}>\mathbf{0}_{|T|}}$ *are given according to*

$$f_i^{\mathcal{K}(\mathbf{s})}(x;\mathbf{w_s},\mathbf{x_s}) = \sum_{\mathbf{n}>\mathbf{0}_{|T|}} a_{\mathbf{n}}\prod_{j\in T}n_j!\sum_{\substack{\mathbf{m}\geq\mathbf{1}_{|\mathbf{s}_j|}\\|\mathbf{m}|=n_j}}\prod_{c\in\mathbf{s}_j}\frac{(w_cx_c)^{m_c}}{m_c!}.$$

□

The following example illustrates how an oracle component for multi-type networks can have the form of Example 5.2.24 while being constructed in a more natural manner.

Example 5.2.25. *Consider multi-type networks such that*

$$\hat{f}_i(x;\mathbf{w_s},\mathbf{x_s}) = f_i^{\mathbf{0}}(x) + F\left(\sum_{j\in T}F_j\left(\sum_{c\in\mathbf{s}_j}w_cx_c\right)\right).$$

with $F(X)=\sum_{n=1}^{N} a_n X^n$ *and* $F_j(X)=\sum_{n=1}^{N_j} a_n^j X^n$, *for all* $j\in T$. *Then, we have that*

$$\begin{aligned} F\left(\sum_{j\in T} F_j\left(\sum_{c\in\mathbf{s}_j} w_c x_c\right)\right) &= \sum_{n=1}^{N} a_n \left(\sum_{j\in T} F_j\left(\sum_{c\in\mathbf{s}_j} w_c x_c\right)\right)^n \\ &= \sum_{n=1}^{N} a_n \sum_{\substack{\mathbf{m}\geq\mathbf{0}_{|T|}\\|\mathbf{m}|=n}} \binom{n}{\mathbf{m}} \prod_{j\in T} F_j\left(\sum_{c\in\mathbf{s}_j} w_c x_c\right)^{m_j} \\ &= \sum_{n=1}^{N} a_n n! \sum_{\substack{\mathbf{m}\geq\mathbf{0}_{|T|}\\|\mathbf{m}|=n}} \prod_{j\in T} \frac{1}{m_j!} F_j\left(\sum_{c\in\mathbf{s}_j} w_c x_c\right)^{m_j} \\ &= \sum_{n=1}^{N} a_n n! \sum_{\substack{\mathbf{m}\geq\mathbf{0}_{|T|}\\|\mathbf{m}|=n}} \prod_{j\in T} \frac{1}{m_j!} \left(\sum_{l=1}^{N_j} a_l^j \left(\sum_{c\in\mathbf{s}_j} w_c x_c\right)^l\right)^{m_j}. \end{aligned}$$

Note that the product of polynomials can be obtained by the convolution of their coefficients. Therefore, raising a polynomial to the power n is equivalent to convolving its coefficients with themselves n times. In particular, $(\sum_{n=1}^{N} a_n X^n)^m$ *can be written as* $\sum_{n=m}^{Nm} b_n X^n$, *with*

$$b_n = \sum_{\substack{\mathbf{l}\geq\mathbf{1}_m\\ \mathbf{l}\leq N\,\mathbf{1}_m\\ |\mathbf{l}|=n}} \prod_{i=1}^{m} a_{l_i}.$$

Therefore, for any fixed $\mathbf{n}>\mathbf{0}_{|T|}$, *the coefficient* $a_\mathbf{n}$ *associated with* $\prod_{j\in T}\left(\sum_{c\in\mathbf{s}_j} w_c x_c\right)^{n_j}$ *as in Example 5.2.24 is given by*

$$a_\mathbf{n} = \sum_{n=1}^{N} a_n n! \sum_{\substack{\mathbf{m}\geq\mathbf{0}_{|T|}\\|\mathbf{m}|=n}} \prod_{j\in T} \left(\frac{1}{m_j!} \sum_{\substack{\mathbf{l}\geq\mathbf{1}_{m_j}\\ \mathbf{l}\leq N_j\,\mathbf{1}_{m_j}\\ |\mathbf{l}|=n_j}} \prod_{i=1}^{m_j} a_{l_i}^j \right).$$

Note that the outer function F is the one responsible for the existence of non-zero coupling components with mixed typing. Consider, for instance, $N=1$. *Then,* $a_\mathbf{n}$ *with* $\mathbf{n}>\mathbf{0}_{|T|}$ *can only be non-zero whenever* $\mathbf{n}=a\,\mathbf{e}_j$ *for some* $j\in T$. *The reason is that the only way for the innermost sum to be non-zero whenever* $m_j=0$, *is for* n_j *to be zero as well. In that situation, we have a sum over one valid index (the* 0*-tuple) of an empty product, which results in* 1. *Similarly, if we consider* $N=2$, *then,* $a_\mathbf{n}$ *with* $\mathbf{n}>\mathbf{0}_{|T|}$ *can only be non-zero whenever* $\mathbf{n}=a\,\mathbf{e}_j+b\,\mathbf{e}_k$ *for some* $j,k\in T$, *and so on.* □

We now introduce the second composition scheme.

5.3 Decomposition into basis components

In Section 5.2 we introduced a scheme that decomposes any given oracle component $\hat{f}_i \in \hat{\mathcal{F}}_i$ into a family of coupling components $\{f_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}}$ that have the properties described in Theorem 5.2.7. With that, one can easily verify in a very systematic way if some function $\hat{f}_i$ that is used to model the behavior of cells in a network satisfies the properties given by Definition 2.4.1.

Although this decomposition works well for verification, it is lacking from the perspective of design. The reason for this is the item 2 of Theorem 5.2.7. It forces all coupling components $\{f_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}}$ to be interdependent. Therefore, it is not clear at all what are exactly the degrees of freedom that are available for us, nor how one would even start when choosing such functions.

In this section, we use the previous decomposition as an essential stepping stone in order to create another with more desirable properties.

For that, we require the use of the multiplicity notation and also Stirling numbers of the first and second kinds, which we now describe.

5.3.1 Multiplicity notation

We now introduce the multiplicity notation, which simplifies the following work.

By $\mathbf{m}\mathbf{w}_\mathbf{s}$, with $\mathbf{m} \geq \mathbf{0}_{|\mathbf{s}|}$, we mean that each entry w_c of the vector $\mathbf{w}_\mathbf{s}$ is expanded into m_c entries of the same value. For instance, consider

$$\mathbf{w}_\mathbf{s} = \begin{bmatrix} w_a \\ w_b \end{bmatrix}, \quad \mathbf{m} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}. \qquad \text{Then, } \mathbf{m}\mathbf{w}_\mathbf{s} = \begin{bmatrix} w_a \\ w_b \\ w_b \end{bmatrix}.$$

Note that the number of elements in the resulting vector $\mathbf{m}\mathbf{w}_\mathbf{s}$ is $|\mathbf{m}\mathbf{s}| = |\mathbf{m}|$, which in this case is 3. Moreover, multiplicities can be composed. That is, we can apply some $\overline{\mathbf{m}}$ to the previous $\mathbf{m}\mathbf{w}_\mathbf{s}$, in order to obtain $\overline{\mathbf{m}}\mathbf{m}\mathbf{w}_\mathbf{s}$, which requires $\overline{\mathbf{m}} \geq \mathbf{0}_{|\mathbf{m}|}$. For instance, we could have

$$\overline{\mathbf{m}} = \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix}, \quad \overline{\mathbf{m}}\mathbf{m}\mathbf{w}_\mathbf{s} = \begin{bmatrix} w_a \\ w_a \\ \hline w_b \\ \hline w_b \\ w_b \end{bmatrix},$$

where the horizontal bars are just for illustration purposes in order to make the expansion of $\mathbf{m}\mathbf{w}_\mathbf{s}$ into $\overline{\mathbf{m}}\mathbf{m}\mathbf{w}_\mathbf{s}$ clearer. Note that $|\overline{\mathbf{m}}\mathbf{m}\mathbf{s}| = |\overline{\mathbf{m}}| = 5$. Moreover, applying the successive multiplicities

($\overline{\mathbf{m}}$ after $\mathbf{m}$) is equivalent to applying a single multiplicity $\mathbf{M}$, in our case, we have,

$$\mathbf{M} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}, \quad \mathbf{M}\mathbf{w_s} = \left[\begin{array}{c} w_a \\ w_a \\ \hline w_b \\ w_b \\ w_b \end{array}\right].$$

Note that $|\mathbf{M}| = |\overline{\mathbf{m}}| = 5$. We say that $\mathbf{M} = \overline{\mathbf{m}}\mathbf{m}$, where $\overline{\mathbf{m}}\mathbf{m}$ is the composition of the two multiplicities $\overline{\mathbf{m}}$ and $\mathbf{m}$. This should not be confused with extending $\mathbf{m}$ according to $\overline{\mathbf{m}}$, which has a completely different meaning. In our example, we have

$$\overline{\mathbf{m}}\mathbf{m} = \left[\begin{array}{c} 2 \\ \hline 1 \\ 2 \end{array}\right] \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \mathbf{M}.$$

The $|\mathbf{m}|$ entries of $\overline{\mathbf{m}}$ can be divided according to the values of $\mathbf{m}$, which in this case is a first block with one element and a second block with two elements. Note that each block will affect a different element of the original vector we are applying $\overline{\mathbf{m}}\mathbf{m}$ to (e.g., $\mathbf{w_s}$), that is, each element of the i^{th} block of $\overline{\mathbf{m}}$ expands the i^{th} element of $\mathbf{w_s}$ that amount of times. In conclusion, to find the equivalent multiplicity $\mathbf{M}$ we just need to sum each block of the multiplicity $\overline{\mathbf{m}}$, in which, the blocks are defined according to $\mathbf{m}$.

5.3.2 Stirling numbers

The Stirling numbers of the first and second kinds are integers that appear in combinatorics, in particular when studying partitions and permutations Comtet (1974). In this section we will define them with respect to their recurrence relations. The related results that will be used in the sections are presented and proved in Appendix A.1.

Definition 5.3.1. *The unsigned Stirling numbers of the first kind, $\mathcal{S}_1(n,k)$, with $n,k \geq 0$, are given by the recurrence relation*

$$\mathcal{S}_1(n,k) = (n-1)\mathcal{S}_1(n-1,k) + \mathcal{S}_1(n-1,k-1), \quad n,k > 0,$$

together with the boundary conditions

$$\begin{aligned} \mathcal{S}_1(0,0) &= 1, \\ \mathcal{S}_1(0,k) &= 0, \quad k > 0, \\ \mathcal{S}_1(n,0) &= 0, \quad n > 0. \end{aligned}$$

□

Definition 5.3.2. *The Stirling numbers of the second kind, $\mathcal{S}_2(n,k)$, with $n,k \geq 0$, are given by the recurrence relation*

$$\mathcal{S}_2(n,k) = k\mathcal{S}_2(n-1,k) + \mathcal{S}_2(n-1,k-1), \quad n,k > 0,$$

together with the boundary conditions

$$\begin{aligned}\mathcal{S}_2(0,0) &= 1, \\ \mathcal{S}_2(0,k) &= 0, \quad k > 0, \\ \mathcal{S}_2(n,0) &= 0, \quad n > 0.\end{aligned}$$

□

5.3.3 Finite coupling order

We denote by $\hat{\mathcal{F}}_i^{<\infty}$ the subset of elements in $\hat{\mathcal{F}}_i$ whose set of coupling components $\{f_i^{\mathbf{k}}\}_{\mathbf{k} \geq \mathbf{0}_{|T|}}$ has only finitely many non-zero terms. From Lemma 5.2.9, this forms a subspace. We now show that we can represent the elements of $\hat{\mathcal{F}}_i^{<\infty}$ by a set of functions $\{{}^b f_i^{\mathbf{k}}\}_{\mathbf{k} \geq \mathbf{0}_{|T|}}$, called **basis components**, which have simpler properties than the coupling components $\{f_i^{\mathbf{k}}\}_{\mathbf{k} \geq \mathbf{0}_{|T|}}$. In particular, they are decoupled from one another. These functions have the following structure.

Definition 5.3.3. *A* ***basis component*** ${}^b f_i^{\mathbf{k}}$*, with* $\mathbf{k} \geq \mathbf{0}_{|T|}$*, is a function defined on*

$${}^b f_i^{\mathbf{k}} \colon \mathbb{X}_i \times \mathcal{M}_i^{\mathbf{k}} \times \mathbb{X}^{\mathbf{k}} \to \mathbb{Y}_i, \tag{5.14}$$

such that:

1. *If σ is any* ***permutation*** *matrix (of appropriate dimension), then*

$${}^b f_i^{\mathbf{k}}(x; \mathbf{w}, \mathbf{x}) = {}^b f_i^{\mathbf{k}}(x; \sigma\mathbf{w}, \sigma\mathbf{x}). \tag{5.15}$$

2. *If $k_j > 0$, then ${}^b f_i^{\mathbf{k}}$ is* ***additive in the weights*** *with respect to type j. That is,*

$${}^b f_i^{\mathbf{k}}\left(x; \begin{bmatrix} w_{j_1} \| w_{j_2} \\ \mathbf{w}_{\bar{\mathbf{s}}} \end{bmatrix}, \begin{bmatrix} x_{j_{12}} \\ \mathbf{x}_{\bar{\mathbf{s}}} \end{bmatrix}\right) = {}^b f_i^{\mathbf{k}}\left(x; \begin{bmatrix} w_{j_1} \\ \mathbf{w}_{\bar{\mathbf{s}}} \end{bmatrix}, \begin{bmatrix} x_{j_{12}} \\ \mathbf{x}_{\bar{\mathbf{s}}} \end{bmatrix}\right) + {}^b f_i^{\mathbf{k}}\left(x; \begin{bmatrix} w_{j_2} \\ \mathbf{w}_{\bar{\mathbf{s}}} \end{bmatrix}, \begin{bmatrix} x_{j_{12}} \\ \mathbf{x}_{\bar{\mathbf{s}}} \end{bmatrix}\right). \tag{5.16}$$

□

Corollary 5.3.4. *Given a basis component ${}^b f_i^{\mathbf{k}}$, if* ***any*** *of the entries of $\mathbf{w}$ is 0_{ij} for some $j \in T$, then*

$${}^b f_i^{\mathbf{k}}(x; \mathbf{w}, \mathbf{x}) = 0_{\mathbb{Y}_i}. \tag{5.17}$$

□

Proof. From item 2 of Definition 5.3.3, we know that

$$
{}^{b}f_i^{\mathbf{k}}\left(x;\begin{bmatrix} w_{j_1}\|0_{ij}\\ \mathbf{w}_{\bar{\mathbf{s}}}\end{bmatrix},\begin{bmatrix} x_{j_{12}}\\ \mathbf{x}_{\bar{\mathbf{s}}}\end{bmatrix}\right)={}^{b}f_i^{\mathbf{k}}\left(x;\begin{bmatrix} w_{j_1}\\ \mathbf{w}_{\bar{\mathbf{s}}}\end{bmatrix},\begin{bmatrix} x_{j_{12}}\\ \mathbf{x}_{\bar{\mathbf{s}}}\end{bmatrix}\right)+{}^{b}f_i^{\mathbf{k}}\left(x;\begin{bmatrix} 0_{ij}\\ \mathbf{w}_{\bar{\mathbf{s}}}\end{bmatrix},\begin{bmatrix} x_{j_{12}}\\ \mathbf{x}_{\bar{\mathbf{s}}}\end{bmatrix}\right).
$$

Since $w_{j_1}\|0_{ij}=w_{j_1}$, this implies that

$$
{}^{b}f_i^{\mathbf{k}}\left(x;\begin{bmatrix} 0_{ij}\\ \mathbf{w}_{\bar{\mathbf{s}}}\end{bmatrix},\begin{bmatrix} x_{j_{12}}\\ \mathbf{x}_{\bar{\mathbf{s}}}\end{bmatrix}\right)=0_{\mathbb{Y}_i}.
$$

The fact that this applies to every $j\in T$, together with item 1 of Definition 5.3.3, proves the result for a zero in any entry of $\mathbf{w}$. ■

The following result assigns the appropriate basis components $\{{}^{b}f_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}}$ to the elements of $\hat{\mathcal{F}}_i^{<\infty}$ by relating them, bijectively, to the coupling components $\{f_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}}$. We now use the following shorthand notation

$$
f_i^{\mathcal{K}(\mathbf{ms})}(x;\mathbf{m},\mathbf{w_s},\mathbf{x_s}):=f_i^{\mathcal{K}(\mathbf{ms})}(x;\mathbf{mw_s},\mathbf{mx_s}).
$$

Theorem 5.3.5. *Assuming the related set $\mathbb{Y}_i$ to be a vector space, there is a bijection between the set of coupling components $\{f_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}}$ of elements in $\hat{\mathcal{F}}_i^{<\infty}$, and the set of basis components $\{{}^{b}f_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}}$ with finitely many non-zero terms. In particular, this bijection is given by the following equivalent expressions,*

$$
f_i^{\mathcal{K}(\mathbf{s})}(x;\mathbf{w_s},\mathbf{x_s})=\sum_{\mathbf{m}\geq\mathbf{1}_{|\mathbf{s}|}}\frac{1}{\prod_{c\in\mathbf{s}}m_c!}\,{}^{b}f_i^{\mathcal{K}(\mathbf{ms})}(x;\mathbf{m},\mathbf{w_s},\mathbf{x_s}),\tag{5.18}
$$

$$
{}^{b}f_i^{\mathcal{K}(\mathbf{s})}(x;\mathbf{w_s},\mathbf{x_s})=\sum_{\mathbf{m}\geq\mathbf{1}_{|\mathbf{s}|}}\frac{(-1)^{|\mathbf{m}|-|\mathbf{s}|}}{\prod_{c\in\mathbf{s}}m_c}f_i^{\mathcal{K}(\mathbf{ms})}(x;\mathbf{m},\mathbf{w_s},\mathbf{x_s}),\tag{5.19}
$$

and in general for multiplicities $\mathbf{m}\geq\mathbf{0}_{|\mathbf{s}|}$,

$$
f_i^{\mathcal{K}(\mathbf{ms})}(x;\mathbf{m},\mathbf{w_s},\mathbf{x_s})=\sum_{\mathbf{M}\geq\mathbf{0}_{|\mathbf{s}|}}\left(\prod_{c\in\mathbf{s}}\frac{m_c!}{M_c!}\mathcal{S}_2(M_c,m_c)\right){}^{b}f_i^{\mathcal{K}(\mathbf{Ms})}(x;\mathbf{M},\mathbf{w_s},\mathbf{x_s}),\tag{5.20}
$$

$$
{}^{b}f_i^{\mathcal{K}(\mathbf{ms})}(x;\mathbf{m},\mathbf{w_s},\mathbf{x_s})=\sum_{\mathbf{M}\geq\mathbf{0}_{|\mathbf{s}|}}(-1)^{|\mathbf{M}|-|\mathbf{m}|}\left(\prod_{c\in\mathbf{s}}\frac{m_c!}{M_c!}\mathcal{S}_1(M_c,m_c)\right)f_i^{\mathcal{K}(\mathbf{Ms})}(x;\mathbf{M},\mathbf{w_s},\mathbf{x_s}).\tag{5.21}
$$

□

In order to prove this, we require Lemmas 5.3.6 to 5.3.12, which are proven in Appendix A.1.

Lemma 5.3.6. *For* $\mathbf{m}, \mathbf{M} \geq \mathbf{0}_k$*, with* $k \geq 0$*, we have that*

$$\sum_{\substack{\overline{\mathbf{m}} \geq \mathbf{1}_{|\mathbf{m}|} \\ \overline{\mathbf{m}}\mathbf{m} = \mathbf{M}}} \frac{1}{\prod_{i=1}^{|\mathbf{m}|} \overline{m}_i} = \prod_{i=1}^{k} \frac{m_i!}{M_i!} \mathcal{S}_1(M_i, m_i). \tag{5.22}$$

□

Lemma 5.3.7. *For* $\mathbf{m}, \mathbf{M} \geq \mathbf{0}_k$*, with* $k \geq 0$*, we have that*

$$\sum_{\substack{\overline{\mathbf{m}} \geq \mathbf{1}_{|\mathbf{m}|} \\ \overline{\mathbf{m}}\mathbf{m} = \mathbf{M}}} \frac{1}{\prod_{i=1}^{|\mathbf{m}|} \overline{m}_i!} = \prod_{i=1}^{k} \frac{m_i!}{M_i!} \mathcal{S}_2(M_i, m_i). \tag{5.23}$$

□

Lemma 5.3.8. *For* $\mathbf{M} \geq \mathbf{0}_k$*, with* $k \geq 0$*, we have that*

$$\sum_{\mathbf{m} \geq \mathbf{1}_k} \prod_{i=1}^{k} (-1)^{m_i} \mathcal{S}_1(M_i, m_i) = \begin{cases} (-1)^k & \text{if } \mathbf{M} = \mathbf{1}_k, \\ 0 & \text{otherwise}. \end{cases} \tag{5.24}$$

□

Lemma 5.3.9. *For* $\mathbf{M} \geq \mathbf{0}_k$*, with* $k \geq 0$*, we have that*

$$\sum_{\mathbf{m} \geq \mathbf{1}_k} \prod_{i=1}^{k} (-1)^{m_i} (m_i - 1)! \mathcal{S}_2(M_i, m_i) = \begin{cases} (-1)^k & \text{if } \mathbf{M} = \mathbf{1}_k, \\ 0 & \text{otherwise}. \end{cases} \tag{5.25}$$

□

Lemma 5.3.10. *Consider a function* ${}^b f_i^{\mathbf{k}}$ *with the properties in Definition 5.3.3, for some* $\mathbf{k} \geq \mathbf{0}_{|T|}$*. For every* $m_{12} \geq 0, \overline{\mathbf{m}} \geq \mathbf{0}_{|\overline{\mathbf{s}}|}$*, such that* $\mathbf{k} = \mathcal{K}(\overline{\mathbf{m}}\overline{\mathbf{s}}) + m_{12}\,\mathbf{e}_j$*, we have that*

$$ {}^b f_i^{\mathbf{k}}\left(x; \begin{bmatrix} m_{12} \\ \overline{\mathbf{m}} \end{bmatrix}, \begin{bmatrix} w_{j_1} \| w_{j_2} \\ \mathbf{w}_{\overline{\mathbf{s}}} \end{bmatrix}, \begin{bmatrix} x_{j_{12}} \\ \mathbf{x}_{\overline{\mathbf{s}}} \end{bmatrix}\right) = \sum_{\substack{m_1, m_2 \geq 0 \\ m_1 + m_2 = m_{12}}} \binom{m_{12}}{m_1, m_2} {}^b f_i^{\mathbf{k}}\left(x; \begin{bmatrix} m_1 \\ m_2 \\ \overline{\mathbf{m}} \end{bmatrix}, \begin{bmatrix} w_{j_1} \\ w_{j_2} \\ \mathbf{w}_{\overline{\mathbf{s}}} \end{bmatrix}, \begin{bmatrix} x_{j_{12}} \\ x_{j_{12}} \\ \mathbf{x}_{\overline{\mathbf{s}}} \end{bmatrix}\right). \tag{5.26}$$

□

Lemma 5.3.11. *Consider a family of functions* $\{f_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}}$ *with the properties in Theorem 5.2.7, for some* $\mathbf{k}\geq\mathbf{0}_{|T|}$. *For every* $m_{12}\geq 0, \overline{\mathbf{m}}\geq\mathbf{0}_{|\overline{\mathbf{s}}|}$, *such that* $\overline{\mathbf{k}}=\mathcal{K}(\overline{\mathbf{m}}\overline{\mathbf{s}})$, *we have that*

$$f_i^{\overline{\mathbf{k}}+m_{12}\mathbf{e}_j}\left(x;\begin{bmatrix}m_{12}\\ \overline{\mathbf{m}}\end{bmatrix},\begin{bmatrix}w_{j_1}\|w_{j_2}\\ \mathbf{w}_{\overline{\mathbf{s}}}\end{bmatrix},\begin{bmatrix}x_{j_{12}}\\ \mathbf{x}_{\overline{\mathbf{s}}}\end{bmatrix}\right) \tag{5.27}$$

$$=\sum_{\substack{m_1,m_2\geq 0\\ m_1,m_2\leq m_{12}\\ m_1+m_2\geq m_{12}}} B(m_1,m_2,m_{12})f_i^{\overline{\mathbf{k}}+(m_1+m_2)\mathbf{e}_j}\left(x;\begin{bmatrix}m_1\\ m_2\\ \overline{\mathbf{m}}\end{bmatrix},\begin{bmatrix}w_{j_1}\\ w_{j_2}\\ \mathbf{w}_{\overline{\mathbf{s}}}\end{bmatrix},\begin{bmatrix}x_{j_{12}}\\ x_{j_{12}}\\ \mathbf{x}_{\overline{\mathbf{s}}}\end{bmatrix}\right),$$

where $B(m_1,m_2,m_{12})$ *is defined as*

$$B(m_1,m_2,m_{12}):=\binom{m_{12}}{m_{12}-m_1,m_{12}-m_2,m_1+m_2-m_{12}}.$$

□

Lemma 5.3.12. *For every* $m_1,m_2\in\mathbb{N}_0$, *we have that*

$$\sum_{\substack{n\geq 1,m_1,m_2\\ n\leq m_1+m_2}}\frac{(-1)^n}{n}\binom{n}{n-m_1,n-m_2,m_1+m_2-n}=\begin{cases}\frac{(-1)^{m_1}}{m_1} & \text{if } m_1\geq 1, m_2=0,\\ \frac{(-1)^{m_2}}{m_2} & \text{if } m_1=0, m_2\geq 1,\\ 0 & \text{otherwise.}\end{cases} \tag{5.28}$$

. □

Proof of Theorem 5.3.5. Firstly, we prove that if both $\{f_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}}$ and $\{{}^b f_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}}$ have finitely many non-zero terms, then, Equations (5.18) to (5.21) are all equivalent. Note that the assumption implies that all the sums indexed at $\mathbf{m}\geq\mathbf{1}_{|\mathbf{s}|}$ and $\mathbf{M}\geq\mathbf{0}_{|\mathbf{s}|}$ have finite support. That is, they are actually finite sums in disguise and there are no convergence issues.
We now prove the equivalence of Equations (5.18) to (5.21) by proving the cycle of implications Equation (5.18) $\implies$ Equation (5.20) $\implies$ Equation (5.19) $\implies$ Equation (5.21) $\implies$ Equation (5.18).
Assume Equation (5.18). Direct substitution gives us

$$f_i^{\mathcal{K}(\mathbf{ms})}(x;\mathbf{m},\mathbf{w}_{\mathbf{s}},\mathbf{x}_{\mathbf{s}})=\sum_{\overline{\mathbf{m}}\geq\mathbf{1}_{|\mathbf{m}|}}\frac{1}{\prod_{i=1}^{|\mathbf{m}|}\overline{m}_i!}\,{}^b f_i^{\mathcal{K}(\overline{\mathbf{m}}\mathbf{ms})}(x;\overline{\mathbf{m}}\mathbf{m},\mathbf{w}_{\mathbf{s}},\mathbf{x}_{\mathbf{s}}).$$

Since we are dealing with finite sums, we can freely reorder the terms such that we merge together the pairs $(\overline{\mathbf{m}},\mathbf{m})$ such that $\overline{\mathbf{m}}\mathbf{m}=\mathbf{M}$. That is,

$$f_i^{\mathcal{K}(\mathbf{ms})}(x;\mathbf{m},\mathbf{w}_{\mathbf{s}},\mathbf{x}_{\mathbf{s}})=\sum_{\mathbf{M}\geq\mathbf{0}_{|\mathbf{s}|}}\sum_{\substack{\overline{\mathbf{m}}\geq\mathbf{1}_{|\mathbf{m}|}\\ \overline{\mathbf{m}}\mathbf{m}=\mathbf{M}}}\frac{1}{\prod_{i=1}^{|\mathbf{m}|}\overline{m}_i!}\,{}^b f_i^{\mathcal{K}(\mathbf{Ms})}(x;\mathbf{M},\mathbf{w}_{\mathbf{s}},\mathbf{x}_{\mathbf{s}}),$$

which from Lemma 5.3.7 simplifies into Equation (5.20). Therefore, Equation (5.18) $\implies$ Equation (5.20).
We now assume Equation (5.20). Using this on the right hand side of Equation (5.19) we get

$$\sum_{\mathbf{m}\geq\mathbf{1}_{|\mathbf{s}|}} \frac{(-1)^{|\mathbf{m}|-|\mathbf{s}|}}{\prod_{c\in\mathbf{s}} m_c} f_i^{\mathcal{K}(\mathbf{ms})}(x;\mathbf{m},\mathbf{w_s},\mathbf{x_s}) = \sum_{\mathbf{m}\geq\mathbf{1}_{|\mathbf{s}|}} \frac{(-1)^{|\mathbf{m}|-|\mathbf{s}|}}{\prod_{c\in\mathbf{s}} m_c} \sum_{\mathbf{M}\geq\mathbf{0}_{|\mathbf{s}|}} \left(\prod_{c\in\mathbf{s}} \frac{m_c!}{M_c!}\mathcal{S}_2(M_c,m_c)\right) {}^b f_i^{\mathcal{K}(\mathbf{Ms})}(x;\mathbf{M},\mathbf{w_s},\mathbf{x_s}).$$

Exchanging the order of the two sums and simplifying we get

$$\sum_{\mathbf{M}\geq\mathbf{0}_{|\mathbf{s}|}} \frac{(-1)^{|\mathbf{s}|}}{\prod_{c\in\mathbf{s}} M_c!} \left(\sum_{\mathbf{m}\geq\mathbf{1}_{|\mathbf{s}|}} \prod_{c\in\mathbf{s}} (-1)^{m_c}(m_c-1)!\mathcal{S}_2(M_c,m_c)\right) {}^b f_i^{\mathcal{K}(\mathbf{Ms})}(x;\mathbf{M},\mathbf{w_s},\mathbf{x_s}),$$

which from Lemma 5.3.9 simplifies into ${}^b f_i^{\mathcal{K}(\mathbf{s})}(x;\mathbf{w_s},\mathbf{x_s})$, the left hand side of Equation (5.19). Therefore, Equation (5.20) $\implies$ Equation (5.19).
We now assume Equation (5.19). Direct substitution gives us

$${}^b f_i^{\mathcal{K}(\mathbf{ms})}(x;\mathbf{m},\mathbf{w_s},\mathbf{x_s}) = \sum_{\overline{\mathbf{m}}\geq\mathbf{1}_{|\mathbf{m}|}} \frac{(-1)^{|\overline{\mathbf{m}}|-|\mathbf{m}|}}{\prod_{i=1}^{|\mathbf{m}|} \overline{m}_i} f_i^{\mathcal{K}(\overline{\mathbf{m}}\mathbf{ms})}(x;\overline{\mathbf{m}}\mathbf{m},\mathbf{w_s},\mathbf{x_s}).$$

Merging together the pairs $(\overline{\mathbf{m}},\mathbf{m})$ such that $\overline{\mathbf{m}}\mathbf{m}=\mathbf{M}$, we obtain

$${}^b f_i^{\mathcal{K}(\mathbf{ms})}(x;\mathbf{m},\mathbf{w_s},\mathbf{x_s}) = \sum_{\mathbf{M}\geq\mathbf{0}_{|\mathbf{s}|}} (-1)^{|\mathbf{M}|-|\mathbf{m}|} \sum_{\substack{\overline{\mathbf{m}}\geq\mathbf{1}_{|\mathbf{m}|}\\ \overline{\mathbf{m}}\mathbf{m}=\mathbf{M}}} \frac{1}{\prod_{i=1}^{|\mathbf{m}|} \overline{m}_i} f_i^{\mathcal{K}(\mathbf{Ms})}(x;\mathbf{M},\mathbf{w_s},\mathbf{x_s}).$$

Note that $|\overline{\mathbf{m}}| = |\mathbf{M}|$. From Lemma 5.3.6, this simplifies into Equation (5.21). Therefore, Equation (5.19) $\implies$ Equation (5.21).
Finally, we assume Equation (5.21). Using this on the right hand side of Equation (5.18) we get

$$\sum_{\mathbf{m}\geq\mathbf{1}_{|\mathbf{s}|}} \frac{1}{\prod_{c\in\mathbf{s}} m_c!} {}^b f_i^{\mathcal{K}(\mathbf{ms})}(x;\mathbf{m},\mathbf{w_s},\mathbf{x_s}) = \sum_{\mathbf{m}\geq\mathbf{1}_{|\mathbf{s}|}} \frac{1}{\prod_{c\in\mathbf{s}} m_c!} \sum_{\mathbf{M}\geq\mathbf{0}_{|\mathbf{s}|}} (-1)^{|\mathbf{M}|-|\mathbf{m}|} \left(\prod_{c\in\mathbf{s}} \frac{m_c!}{M_c!}\mathcal{S}_1(M_c,m_c)\right) f_i^{\mathcal{K}(\mathbf{Ms})}(x;\mathbf{M},\mathbf{w_s},\mathbf{x_s}).$$

Exchanging the order of the two sums and simplifying we get

$$\sum_{\mathbf{M}\geq\mathbf{0}_{|\mathbf{s}|}} \frac{(-1)^{|\mathbf{M}|}}{\prod_{c\in\mathbf{s}} M_c!} \left(\sum_{\mathbf{m}\geq\mathbf{1}_{|\mathbf{s}|}} \prod_{c\in\mathbf{s}} (-1)^{m_c}\mathcal{S}_1(M_c,m_c)\right) f_i^{\mathcal{K}(\mathbf{Ms})}(x;\mathbf{M},\mathbf{w_s},\mathbf{x_s}),$$

which from Lemma 5.3.8 simplifies into Equation (5.18). This completes the proof that Equations (5.18) to (5.21) are equivalent under the assumption that both $\{f_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}}$ and $\{{}^b f_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}}$

have finitely many non-zero terms. We now weaken this assumption by showing that one of them having finitely many non-zero terms implies the other also having that property.

Assume $\{f_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}}$ has finitely many terms. Then, there is some $\mathbf{K}\geq\mathbf{0}_{|T|}$ such that all non-zero terms are inside the subset $\{f_i^{\mathbf{k}}\}_{\mathbf{k}\leq\mathbf{K}}$. Note that the sums Equation (5.19) are always finite, which means that the corresponding $\{{}^b f_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}}$ is well-defined. Furthermore, every ${}^b f_i^{\mathbf{k}}$ such that $\mathbf{k}$ does not obey $\mathbf{k}\leq\mathbf{K}$, is given by a sum of zero terms. Therefore, $\{{}^b f_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}}$ also has all of its non-zero terms inside the subset $\{{}^b f_i^{\mathbf{k}}\}_{\mathbf{k}\leq\mathbf{K}}$, which means that it also has finitely many non-zero terms. Then, the previous assumptions are satisfied and consequently Equations (5.18) to (5.21) are equivalent.

The exact same argument applies when starting with some $\{{}^b f_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}}$ that has finitely many non-zero terms and constructing the corresponding $\{f_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}}$ through Equation (5.18).

We now prove that $\{f_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}}$ has the properties in Theorem 5.2.7 if and only if the corresponding $\{{}^b f_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}}$ has the properties in Definition 5.3.3.

Assume some $\{{}^b f_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}}$ has the properties in Definition 5.3.3. Then, from Equation (5.18), we have that for any permutation matrix σ,

$$\begin{aligned}
f_i^{\mathcal{K}(\mathbf{s})}(x;\mathbf{w_s},\mathbf{x_s}) &= \sum_{\mathbf{m}\geq\mathbf{1}_{|\mathbf{s}|}} \frac{1}{\prod_{c\in\mathbf{s}} m_c!}\,{}^b f_i^{\mathcal{K}(\mathbf{ms})}(x;\mathbf{m},\mathbf{w_s},\mathbf{x_s}) \\
&= \sum_{\mathbf{m}\geq\mathbf{1}_{|\mathbf{s}|}} \frac{1}{\prod_{c\in\mathbf{s}} m_c!}\,{}^b f_i^{\mathcal{K}([\sigma\mathbf{m}]\sigma\mathbf{s})}(x;\sigma\mathbf{m},\sigma\mathbf{w_s},\sigma\mathbf{x_s}) \\
&= \sum_{\overline{\mathbf{m}}\geq\mathbf{1}_{|\mathbf{s}|}} \frac{1}{\prod_{c\in\mathbf{s}} \overline{m}_c!}\,{}^b f_i^{\mathcal{K}(\overline{\mathbf{m}}\sigma\mathbf{s})}(x;\overline{\mathbf{m}},\sigma\mathbf{w_s},\sigma\mathbf{x_s}) \\
&= f_i^{\mathcal{K}(\mathbf{s})}(x;\sigma\mathbf{w_s},\sigma\mathbf{x_s}),
\end{aligned}$$

where $\overline{\mathbf{m}}=\sigma\mathbf{m}$ establishes a bijection between the sets of indexes $\mathbf{m}\geq\mathbf{1}_{|\mathbf{s}|}$ and $\overline{\mathbf{m}}\geq\mathbf{1}_{|\mathbf{s}|}$. Therefore, $\{f_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}}$ satisfies item 1 of Theorem 5.2.7.

Again from Equation (5.18), we have that

$$f_i^{\mathcal{K}(\mathbf{s})}\left(x;\begin{bmatrix} w_{j_1}\|w_{j_2} \\ \mathbf{w}_{\overline{\mathbf{s}}}\end{bmatrix},\mathbf{x_s}\right) = \sum_{\mathbf{m}\geq\mathbf{1}_{|\mathbf{s}|}} \frac{1}{\prod_{c\in\mathbf{s}} m_c!}\,{}^b f_i^{\mathcal{K}(\mathbf{ms})}\left(x;\mathbf{m},\begin{bmatrix} w_{j_1}\|w_{j_2} \\ \mathbf{w}_{\overline{\mathbf{s}}}\end{bmatrix},\mathbf{x_s}\right),$$

with $\mathbf{m}=\begin{bmatrix} m_{12} \\ \overline{\mathbf{m}}\end{bmatrix}$ and $\mathbf{x_s}=\begin{bmatrix} x_{j_{12}} \\ \mathbf{x}_{\overline{\mathbf{s}}}\end{bmatrix}$. Applying Lemma 5.3.10, the right hand side expands into

$$\sum_{\substack{m_{12}\geq 1\\ \overline{\mathbf{m}}\geq\mathbf{1}_{|\overline{\mathbf{s}}|}}} \frac{1}{m_{12}!\prod_{c\in\overline{\mathbf{s}}}\overline{m}_c!} \sum_{\substack{m_1,m_2\geq 0\\ m_1+m_2=m_{12}}} \frac{m_{12}!}{m_1!m_2!}\,{}^b f_i^{\mathbf{k}}\left(x;\begin{bmatrix} m_1\\ m_2\\ \overline{\mathbf{m}}\end{bmatrix},\begin{bmatrix} w_{j_1}\\ w_{j_2}\\ \mathbf{w}_{\overline{\mathbf{s}}}\end{bmatrix},\begin{bmatrix} x_{j_{12}}\\ x_{j_{12}}\\ \mathbf{x}_{\overline{\mathbf{s}}}\end{bmatrix}\right).$$

We cancel the $m_{12}!$ terms and merge the two sums, which simplifies the expression into

$$\sum_{\substack{m_1,m_2\geq 0\\ m_1+m_2\geq 1\\ \overline{\mathbf{m}}\geq \mathbf{1}_{|\overline{\mathbf{s}}|}}} \frac{1}{m_1!m_2!\prod_{c\in\overline{\mathbf{s}}}\overline{m}_c!}\,{}^{b}f_i^{\mathbf{k}}\left(x;\begin{bmatrix} m_1\\ m_2\\ \overline{\mathbf{m}}\end{bmatrix},\begin{bmatrix} w_{j_1}\\ w_{j_2}\\ \mathbf{w}_{\overline{\mathbf{s}}}\end{bmatrix},\begin{bmatrix} x_{j_{12}}\\ x_{j_{12}}\\ \mathbf{x}_{\overline{\mathbf{s}}}\end{bmatrix}\right).$$

We now split this sum into three parts. The first with $m_1\geq 1, m_2=0$, the second with $m_1=0, m_2\geq 1$ and the third with $m_1,m_2\geq 1$. Applying Equation (5.18) again, gives us the three terms of the right hand side of item 2 of Theorem 5.2.7.

Finally, consider that some entry of $\mathbf{w}$ is 0_{ij} for some $j\in T$. Then, from Corollary 5.3.4, every term of the sum Equation (5.18) is zero, which means that $\{f_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}}$ satisfies item 3 of Theorem 5.2.7.

We now prove the converse direction. Assume some $\{f_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}}$ has the properties in Theorem 5.2.7. Then, from Equation (5.19), we have that for any permutation matrix σ,

$$\begin{aligned}
{}^{b}f_i^{\mathcal{K}(\mathbf{s})}(x;\mathbf{w}_{\mathbf{s}},\mathbf{x}_{\mathbf{s}}) &= \sum_{\mathbf{m}\geq\mathbf{1}_{|\mathbf{s}|}} \frac{(-1)^{|\mathbf{m}|-|\mathbf{s}|}}{\prod_{c\in\mathbf{s}} m_c} f_i^{\mathcal{K}(\mathbf{ms})}(x;\mathbf{m},\mathbf{w}_{\mathbf{s}},\mathbf{x}_{\mathbf{s}})\\
&= \sum_{\mathbf{m}\geq\mathbf{1}_{|\mathbf{s}|}} \frac{(-1)^{|\mathbf{m}|-|\mathbf{s}|}}{\prod_{c\in\mathbf{s}} m_c} f_i^{\mathcal{K}([\sigma\mathbf{m}]\sigma\mathbf{s})}(x;\sigma\mathbf{m},\sigma\mathbf{w}_{\mathbf{s}},\sigma\mathbf{x}_{\mathbf{s}})\\
&= \sum_{\overline{\mathbf{m}}\geq\mathbf{1}_{|\mathbf{s}|}} \frac{(-1)^{|\overline{\mathbf{m}}|-|\mathbf{s}|}}{\prod_{c\in\mathbf{s}} \overline{m}_c} f_i^{\mathcal{K}(\overline{\mathbf{m}}\sigma\mathbf{s})}(x;\overline{\mathbf{m}},\sigma\mathbf{w}_{\mathbf{s}},\sigma\mathbf{x}_{\mathbf{s}})\\
&= {}^{b}f_i^{\mathcal{K}(\mathbf{s})}(x;\sigma\mathbf{w}_{\mathbf{s}},\sigma\mathbf{x}_{\mathbf{s}}),
\end{aligned}$$

where $\overline{\mathbf{m}}=\sigma\mathbf{m}$ establishes a bijection between the sets of indexes $\mathbf{m}\geq\mathbf{1}_{|\mathbf{s}|}$ and $\overline{\mathbf{m}}\geq\mathbf{1}_{|\mathbf{s}|}$. Therefore, $\{{}^{b}f_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}}$ satisfies item 1 of Definition 5.3.3.

Finally, we have that

$${}^{b}f_i^{\mathcal{K}(\mathbf{s})}\left(x;\begin{bmatrix} w_{j_1}\|w_{j_2}\\ \mathbf{w}_{\overline{\mathbf{s}}}\end{bmatrix},\mathbf{x}_{\mathbf{s}}\right) = \sum_{\mathbf{m}\geq\mathbf{1}_{|\mathbf{s}|}} \frac{(-1)^{|\mathbf{m}|-|\mathbf{s}|}}{\prod_{c\in\mathbf{s}} m_c} f_i^{\mathcal{K}(\mathbf{ms})}\left(x;\mathbf{m},\begin{bmatrix} w_{j_1}\|w_{j_2}\\ \mathbf{w}_{\overline{\mathbf{s}}}\end{bmatrix},\mathbf{x}_{\mathbf{s}}\right),$$

with $\mathbf{m}=\begin{bmatrix} m_{12}\\ \overline{\mathbf{m}}\end{bmatrix}$ and $\mathbf{x}_{\mathbf{s}}=\begin{bmatrix} x_{j_{12}}\\ \mathbf{x}_{\overline{\mathbf{s}}}\end{bmatrix}$. Applying Lemma 5.3.11, the right hand side expands into

$$\sum_{\substack{m_{12}\geq 1\\ \overline{\mathbf{m}}\geq\mathbf{1}_{|\overline{\mathbf{s}}|}}} \frac{(-1)^{|\overline{\mathbf{m}}|-|\mathbf{s}|}}{\prod_{c\in\overline{\mathbf{s}}} m_c}\frac{(-1)^{m_{12}}}{m_{12}} \sum_{\substack{m_1,m_2\geq 0\\ m_1,m_2\leq m_{12}\\ m_1+m_2\geq m_{12}}} B(m_1,m_2,m_{12}) f_i^{\mathcal{K}(\overline{\mathbf{m}}\mathbf{s})+(m_1+m_2)\mathbf{e}_j}\left(x;\begin{bmatrix} m_1\\ m_2\\ \overline{\mathbf{m}}\end{bmatrix},\begin{bmatrix} w_{j_1}\\ w_{j_2}\\ \mathbf{w}_{\overline{\mathbf{s}}}\end{bmatrix},\begin{bmatrix} x_{j_{12}}\\ x_{j_{12}}\\ \mathbf{x}_{\overline{\mathbf{s}}}\end{bmatrix}\right),$$

with $B(m_1,m_2,m_{12})$ as defined in Lemma 5.3.11. Note that we are summing over all tuples $(m_{12},m_1,m_2,\overline{\mathbf{m}})$ with $m_{12}\geq 1$, $m_1,m_2\geq 0$ and $\overline{\mathbf{m}}\geq\mathbf{1}_{|\overline{\mathbf{s}}|}$, such that $m_1,m_2\leq m_{12}$ and $m_1+m_2\geq m_{12}$.

We can then rearrange the two sums into

$$\sum_{\substack{m_1,m_2\geq 0\\ \overline{\mathbf{m}}\geq \mathbf{1}_{|\overline{\mathbf{s}}|}}} \frac{(-1)^{|\overline{\mathbf{m}}|-|\mathbf{s}|}}{\prod_{c\in\overline{\mathbf{s}}} m_c} \left[\sum_{\substack{m_{12}\geq 1, m_1, m_2\\ m_{12}\leq m_1+m_2}} \frac{(-1)^{m_{12}}}{m_{12}} B(m_1,m_2,m_{12})\right] f_i^{\mathcal{K}(\overline{\mathbf{m}}\overline{\mathbf{s}})+(m_1+m_2)\mathbf{e}_j}\left(x;\begin{bmatrix} m_1\\ m_2\\ \overline{\mathbf{m}}\end{bmatrix},\begin{bmatrix} w_{j_1}\\ w_{j_2}\\ \mathbf{w}_{\overline{\mathbf{s}}}\end{bmatrix},\begin{bmatrix} x_{j_{12}}\\ x_{j_{12}}\\ \mathbf{x}_{\overline{\mathbf{s}}}\end{bmatrix}\right).$$

From Lemma 5.3.12, this simplifies into

$$\begin{aligned}
&\sum_{\substack{m_1\geq 1\\ \overline{\mathbf{m}}\geq \mathbf{1}_{|\overline{\mathbf{s}}|}}} \frac{(-1)^{|\overline{\mathbf{m}}|+m_1-|\mathbf{s}|}}{m_1\prod_{c\in\overline{\mathbf{s}}} m_c} f_i^{\mathcal{K}(\mathbf{m}_1\mathbf{s})}\left(x;\begin{bmatrix} m_1\\ \overline{\mathbf{m}}\end{bmatrix},\begin{bmatrix} w_{j_1}\\ \mathbf{w}_{\overline{\mathbf{s}}}\end{bmatrix},\mathbf{x}_{\mathbf{s}}\right)\\
&+\sum_{\substack{m_2\geq 1\\ \overline{\mathbf{m}}\geq \mathbf{1}_{|\overline{\mathbf{s}}|}}} \frac{(-1)^{|\overline{\mathbf{m}}|+m_2-|\mathbf{s}|}}{m_2\prod_{c\in\overline{\mathbf{s}}} m_c} f_i^{\mathcal{K}(\mathbf{m}_2\mathbf{s})}\left(x;\begin{bmatrix} m_2\\ \overline{\mathbf{m}}\end{bmatrix},\begin{bmatrix} w_{j_2}\\ \mathbf{w}_{\overline{\mathbf{s}}}\end{bmatrix},\mathbf{x}_{\mathbf{s}}\right)\\
&= {}^b f_i^{\mathcal{K}(\mathbf{s})}\left(x;\begin{bmatrix} w_{j_1}\\ \mathbf{w}_{\overline{\mathbf{s}}}\end{bmatrix},\mathbf{x}_{\mathbf{s}}\right) + {}^b f_i^{\mathcal{K}(\mathbf{s})}\left(x;\begin{bmatrix} w_{j_2}\\ \mathbf{w}_{\overline{\mathbf{s}}}\end{bmatrix},\mathbf{x}_{\mathbf{s}}\right),
\end{aligned}$$

with $\mathbf{m}_1 = \begin{bmatrix} m_1\\ \overline{\mathbf{m}}\end{bmatrix}$ and $\mathbf{m}_2 = \begin{bmatrix} m_2\\ \overline{\mathbf{m}}\end{bmatrix}$, which gives us the right hand side of item 2 of Definition 5.3.3. ■

An evident but important consequence of Theorem 5.3.5 is the following.

Corollary 5.3.13. *Consider a finite order* $\hat{f}_i \in \hat{\mathcal{F}}_i^{<\infty}$*, with coupling components* $\{f_i^{\mathbf{k}}\}_{\mathbf{k}\geq \mathbf{0}_{|T|}}$ *and with basis components* $\{{}^b f_i^{\mathbf{k}}\}_{\mathbf{k}\geq \mathbf{0}_{|T|}}$*. Then,*

$$f_i^{\mathbf{0}} = {}^b f_i^{\mathbf{0}}. \tag{5.29}$$

□

This can be generalized with the help of the following definition.

Definition 5.3.14. *Consider a family of functions* $\{f_i^{\mathbf{k}}\}_{\mathbf{k}\geq \mathbf{0}_{|T|}}$ *defined on* $f_i^{\mathbf{k}} : \mathbb{X}_i \times \mathcal{M}_i^{\mathbf{k}} \times \mathbb{X}^{\mathbf{k}} \to \mathbb{Y}_i$*. We say that a given index* $\mathbf{k} \geq \mathbf{0}_{|T|}$ *is a* ***locally maximal order*** *if* $f_i^{\mathbf{k}} \neq 0_{\mathbb{Y}_i}$ *and* $f_i^{\overline{\mathbf{k}}} = 0_{\mathbb{Y}_i}$ *for all* $\overline{\mathbf{k}} > \mathbf{k}$ *such that* $\mathbf{k}$ *and* $\overline{\mathbf{k}}$ *have zeros in the same entries.* □

Lemma 5.3.15. *Consider a finite order* $\hat{f}_i \in \hat{\mathcal{F}}_i^{<\infty}$*, with coupling components* $\{f_i^{\mathbf{k}}\}_{\mathbf{k}\geq \mathbf{0}_{|T|}}$ *and with basis components* $\{{}^b f_i^{\mathbf{k}}\}_{\mathbf{k}\geq \mathbf{0}_{|T|}}$*.*
A given index $\mathbf{k} \geq \mathbf{0}_{|T|}$ *is a locally maximal order with respect to* $\{f_i^{\mathbf{k}}\}_{\mathbf{k}\geq \mathbf{0}_{|T|}}$ *if and only if it is a locally maximal order with respect to* $\{{}^b f_i^{\mathbf{k}}\}_{\mathbf{k}\geq \mathbf{0}_{|T|}}$*. Furthermore, if* $\mathbf{k} \geq \mathbf{0}_{|T|}$ *is a locally maximal order, then,*

$$f_i^{\mathbf{k}} = {}^b f_i^{\mathbf{k}}. \tag{5.30}$$

□

Proof. Assume $\mathbf{k} \geq \mathbf{0}_{|T|}$ is a locally maximal order with respect to $\{f_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}}$. This implies, from Equation (5.19), that ${}^b f_i^{\bar{\mathbf{k}}} = 0_{\mathbb{Y}_i}$ whenever $\mathbf{k} < \bar{\mathbf{k}}$ and $\mathbf{k}, \bar{\mathbf{k}}$ have zeros in the same entries. Moreover, when $\bar{\mathbf{k}} = \mathbf{k}$, Equation (5.19) simplifies into Equation (5.30).
The exact same reasoning applies in order to prove the converse direction using Equation (5.18).

■

The following result allows us to build any $\hat{f}_i \in \hat{\mathcal{F}}_i^{<\infty}$ directly from the specification of a simple and decoupled family of basis components $\{{}^b f_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}}$.

Theorem 5.3.16. *Every finite order oracle component $\hat{f}_i \in \hat{\mathcal{F}}_i^{<\infty}$, can be directly expressed in terms of its basis components $\{{}^b f_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}}$ according to*

$$\hat{f}_i\left(x; \mathbf{w_s}, \mathbf{x_s}\right) = \sum_{\mathbf{m}\geq\mathbf{0}_{|\mathbf{s}|}} \frac{1}{\prod_{c\in\mathbf{s}} m_c!} {}^b f_i^{\mathcal{K}(\mathbf{ms})}\left(x; \mathbf{m}, \mathbf{w_s}, \mathbf{x_s}\right). \tag{5.31}$$

□

Proof. We plug in Equation (5.18) on Equation (5.5), which gives us

$$\hat{f}_i\left(x; \mathbf{w_s}, \mathbf{x_s}\right) = \sum_{\bar{\mathbf{s}}\subseteq\mathbf{s}} \sum_{\mathbf{m}\geq\mathbf{1}_{|\bar{\mathbf{s}}|}} \frac{1}{\prod_{c\in\bar{\mathbf{s}}} m_c!} {}^b f_i^{\mathcal{K}(\mathbf{m}\bar{\mathbf{s}})}\left(x; \mathbf{m}, \mathbf{w}_{\bar{\mathbf{s}}}, \mathbf{x}_{\bar{\mathbf{s}}}\right).$$

The result comes directly from merging the two sums. ■

Similarly to Lemma 5.2.9, we see that the representation on this second decomposition is also component-wise linear.

Lemma 5.3.17. *For two finite order oracle components $\hat{f}_i, \hat{g}_i \in \hat{\mathcal{F}}_i^{<\infty}$ with basis components $\{{}^b f_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}}$ and $\{{}^b g_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}}$ respectively, the basis components of $\hat{h}_i = \alpha\hat{f}_i + \hat{g}_i$ are given by $\{\alpha {}^b f_i^{\mathbf{k}} + {}^b g_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}}$ for any scalar α.* □

Proof. This comes directly from writing the basis components explicitly in terms of the coupling components as in Equation (5.19), together with Lemma 5.2.9.

$$\begin{aligned}
&{}^b h_i^{\mathcal{K}(\mathbf{s})}\left(x; \mathbf{w_s}, \mathbf{x_s}\right) \\
&= \sum_{\mathbf{m}\geq\mathbf{1}_{|\mathbf{s}|}} \frac{(-1)^{|\mathbf{m}|-|\mathbf{s}|}}{\prod_{c\in\mathbf{s}} m_c} \left(\alpha f_i^{\mathcal{K}(\mathbf{ms})} + g_i^{\mathcal{K}(\mathbf{ms})}\right)\left(x; \mathbf{m}, \mathbf{w_s}, \mathbf{x_s}\right) \\
&= \alpha\left(\sum_{\mathbf{m}\geq\mathbf{1}_{|\mathbf{s}|}} \frac{(-1)^{|\mathbf{m}|-|\mathbf{s}|}}{\prod_{c\in\mathbf{s}} m_c} f_i^{\mathcal{K}(\mathbf{ms})}\left(x; \mathbf{m}, \mathbf{w_s}, \mathbf{x_s}\right)\right) + \sum_{\mathbf{m}\geq\mathbf{1}_{|\mathbf{s}|}} \frac{(-1)^{|\mathbf{m}|-|\mathbf{s}|}}{\prod_{c\in\mathbf{s}} m_c} g_i^{\mathcal{K}(\mathbf{ms})}\left(x; \mathbf{m}, \mathbf{w_s}, \mathbf{x_s}\right) \\
&= \alpha {}^b f_i^{\mathcal{K}(\mathbf{s})}\left(x; \mathbf{w_s}, \mathbf{x_s}\right) + {}^b g_i^{\mathcal{K}(\mathbf{s})}\left(x; \mathbf{w_s}, \mathbf{x_s}\right) \\
&= \left(\alpha {}^b f_i^{\mathcal{K}(\mathbf{s})} + {}^b g_i^{\mathcal{K}(\mathbf{s})}\right)\left(x; \mathbf{w_s}, \mathbf{x_s}\right).
\end{aligned}$$

■

The following examples illustrate the proposed decomposition.

Example 5.3.18. *Consider a single-type finite order oracle component $\hat{f}_i \in \hat{\mathcal{F}}_i^{<\infty}$ with basis components $\{{}^b f_i^k\}_{k\geq 0}$ such that, for some fixed $n > 0$*

$$ {}^b f_i^{|\mathbf{s}|}(x; \mathbf{w_s}, \mathbf{x_s}) = \begin{cases} n! \prod_{c\in\mathbf{s}} (w_c x_c) & |\mathbf{s}| = n, \\ f_i^0(x) & |\mathbf{s}| = 0, \\ 0 & \textit{otherwise}. \end{cases} $$

It is clear that $\{{}^b f_i^k\}_{k\geq 0}$ satisfy items 1 and 2 of Definition 5.3.3. Using Theorem 5.3.16 we can find the corresponding oracle component directly. That is,

$$ \begin{aligned} \hat{f}_i(x; \mathbf{w_s}, \mathbf{x_s}) &= \sum_{\mathbf{m}\geq \mathbf{0}_{|\mathbf{s}|}} \frac{1}{\prod_{c\in\mathbf{s}} m_c!} {}^b f_i^{|\mathbf{m}|}(x; \mathbf{m}, \mathbf{w_s}, \mathbf{x_s}) \\ &= f_i^0(x) + n! \sum_{\substack{\mathbf{m}\geq \mathbf{0}_{|\mathbf{s}|} \\ |\mathbf{m}|=n}} \prod_{c\in\mathbf{s}} \frac{(w_c x_c)^{m_c}}{m_c!} \\ &= f_i^0(x) + \left(\sum_{c\in\mathbf{s}} w_c x_c \right)^n, \end{aligned} $$

which is exactly the same oracle component as in Example 5.2.18.
As a sanity check we can easily verify from Equation (5.18) that the coupling components $\{f_i^k\}_{k\geq 0}$ match the previously calculated ones. In particular, for $|\mathbf{s}| > 0$,

$$ f_i^{|\mathbf{s}|}(x; \mathbf{w_s}, \mathbf{x_s}) = \sum_{\mathbf{m}\geq \mathbf{1}_{|\mathbf{s}|}} \frac{1}{\prod_{c\in\mathbf{s}} m_c!} {}^b f_i^{|\mathbf{m}|}(x; \mathbf{m}, \mathbf{w_s}, \mathbf{x_s}) = n! \sum_{\substack{\mathbf{m}\geq \mathbf{1}_{|\mathbf{s}|} \\ |\mathbf{m}|=n}} \prod_{c\in\mathbf{s}} \frac{(w_c x_c)^{m_c}}{m_c!}. $$

Finally, note that Lemma 5.3.15 is verified for $|\mathbf{s}| = n$. That is, $f_i^n = {}^b f_i^n$. □

We now extend Example 5.3.18 to the polynomial case.

Example 5.3.19. *Consider a single-type finite order oracle component $\hat{f}_i \in \hat{\mathcal{F}}_i^{<\infty}$ with basis components $\{{}^b f_i^k\}_{k\geq 0}$ such that, for some fixed $N > 0$*

$$ {}^b f_i^{|\mathbf{s}|}(x; \mathbf{w_s}, \mathbf{x_s}) = \begin{cases} a_{|\mathbf{s}|} |\mathbf{s}|! \prod_{c\in\mathbf{s}} (w_c x_c) & 0 < |\mathbf{s}| \leq N, \\ f_i^0(x) & |\mathbf{s}| = 0, \\ 0 & \textit{otherwise}. \end{cases} $$

It is clear that $\{{}^b f_i^k\}_{k\geq 0}$ satisfy items 1 and 2 of Definition 5.3.3.
From Example 5.3.18 and Lemma 5.3.17, we conclude that the corresponding oracle component

is given by

$$\hat{f}_i(x; \mathbf{w_s}, \mathbf{x_s}) = f_i^0(x) + \sum_{n=1}^{N} a_n \left(\sum_{c \in \mathbf{s}} w_c x_c \right)^n,$$

which is exactly the same oracle component as in Example 5.2.21. Note that we could also obtain this directly through Theorem 5.3.16. □

We now extend the previous result for multi-type networks.

Example 5.3.20. *Consider a multi-type finite order oracle component $\hat{f}_i \in \hat{\mathcal{F}}_i^{<\infty}$ with basis components $\{{}^b f_i^{\mathbf{k}}\}_{\mathbf{k} \geq \mathbf{0}_{|T|}}$ such that, for $\{a_{\mathbf{n}}\}_{\mathbf{n} > \mathbf{0}_{|T|}}$ with finite support,*

$${}^b f_i^{\mathbf{k}}(x; \mathbf{w_s}, \mathbf{x_s}) = \begin{cases} a_{\mathbf{k}} \prod_{j \in T} k_j! \prod_{c \in \mathbf{s}_j} (w_c x_c) & \mathbf{k} > \mathbf{0}_{|T|}, \\ f_i^{\mathbf{0}}(x) & \mathbf{k} = \mathbf{0}_{|T|}. \end{cases}$$

It is clear that $\{{}^b f_i^{\mathbf{k}}\}_{\mathbf{k} \geq \mathbf{0}_{|T|}}$ satisfy items 1 and 2 of Definition 5.3.3. The corresponding oracle component is given by

$$\begin{aligned}
\hat{f}_i(x; \mathbf{w_s}, \mathbf{x_s}) &= \sum_{\mathbf{m} \geq \mathbf{0}_{|\mathbf{s}|}} \frac{1}{\prod_{c \in \mathbf{s}} m_c!} {}^b f_i^{\mathcal{K}(\mathbf{ms})}(x; \mathbf{m}, \mathbf{w_s}, \mathbf{x_s}) \\
&= f_i^{\mathbf{0}}(x) + \sum_{\mathbf{n} > \mathbf{0}_{|T|}} \sum_{\substack{\mathbf{m} \geq \mathbf{0}_{|\mathbf{s}|} \\ |\mathbf{m}_1| = n_1 \\ \cdots \\ |\mathbf{m}_{|T|}| = n_{|T|}}} a_{\mathbf{n}} \left(\prod_{j \in T} n_j! \right) \prod_{c \in \mathbf{s}} \frac{(w_c x_c)^{m_c}}{m_c!} \\
&= f_i^{\mathbf{0}}(x) + \sum_{\mathbf{n} > \mathbf{0}_{|T|}} a_{\mathbf{n}} \prod_{j \in T} \sum_{\substack{\mathbf{m}_j \geq \mathbf{0}_{|\mathbf{s}_j|} \\ |\mathbf{m}_j| = n_j}} n_j! \prod_{c \in \mathbf{s}_j} \frac{(w_c x_c)^{m_c}}{m_c!} \\
&= f_i^{\mathbf{0}}(x) + \sum_{\mathbf{n} > \mathbf{0}_{|T|}} a_{\mathbf{n}} \prod_{j \in T} \left(\sum_{c \in \mathbf{s}_j} w_c x_c \right)^{n_j},
\end{aligned}$$

which is exactly the same oracle component as in Equation (5.14). □

We now consider a slightly more complicated type of basis components.

Example 5.3.21. *Consider a single-type finite order oracle component $\hat{f}_i \in \hat{\mathcal{F}}_i^{<\infty}$ with basis components $\{{}^b f_i^k\}_{k \geq 0}$ such that, for some fixed n, k with $n \geq k > 0$*

$${}^b f_i^{|\mathbf{s}|}(x; \mathbf{w_s}, \mathbf{x_s}) = \begin{cases} (n-k)!k!(\prod_{c \in \mathbf{s}} w_c) e_k(\mathbf{x_s}) & |\mathbf{s}| = n, \\ f_i^0(x) & |\mathbf{s}| = 0, \\ 0 & \text{otherwise}. \end{cases}$$

where e_k denotes what is called elementary symmetric polynomials. With the multi-index notation this can be written as

$$e_k(\mathbf{x_s}) = \sum_{\substack{\mathbf{q}\geq\mathbf{0}_{|\mathbf{s}|}\\ \mathbf{q}\leq\mathbf{1}_{|\mathbf{s}|}\\ |\mathbf{q}|=k}} \prod_{c\in\mathbf{s}} x_c^{q_c}.$$

It is clear that $\{{}^b f_i^k\}_{k\geq 0}$ satisfy items 1 and 2 of Definition 5.3.3. We show that the oracle components $\{f_i^k\}_{k\geq 0}$ can be found to be

$$\hat{f}_i(x;\mathbf{w_s},\mathbf{x_s}) = f_i^{\mathbf{0}}(x) + \left(\sum_{c\in\mathbf{s}} w_c\right)^{n-k}\left(\sum_{c\in\mathbf{s}} w_c x_c\right)^{k}.$$

To prove this, firstly note that

$$e_k(\mathbf{m}\mathbf{x_s}) = \sum_{\substack{\mathbf{q}\geq\mathbf{0}_{|\mathbf{s}|}\\ \mathbf{q}\leq\mathbf{m}\\ |\mathbf{q}|=k}} \prod_{c\in\mathbf{s}} x_c^{q_c}\binom{m_c}{q_c}.$$

Using Theorem 5.3.16,

$$\begin{aligned}
\hat{f}_i(x;\mathbf{w_s},\mathbf{x_s}) &= \sum_{\mathbf{m}\geq\mathbf{0}_{|\mathbf{s}|}} \frac{1}{\prod_{c\in\mathbf{s}} m_c!}\,{}^b f_i^{|\mathbf{m}|}(x;\mathbf{m},\mathbf{w_s},\mathbf{x_s})\\
&= f_i^{\mathbf{0}}(x) + \sum_{\substack{\mathbf{m}\geq\mathbf{0}_{|\mathbf{s}|}\\ |\mathbf{m}|=n}} \frac{1}{\prod_{c\in\mathbf{s}} m_c!}\,{}^b f_i^{|\mathbf{m}|}(x;\mathbf{m},\mathbf{w_s},\mathbf{x_s})\\
&= f_i^{\mathbf{0}}(x) + (n-k)!k! \sum_{\substack{\mathbf{m}\geq\mathbf{0}_{|\mathbf{s}|}\\ |\mathbf{m}|=n}} \frac{1}{\prod_{c\in\mathbf{s}} m_c!}\left(\prod_{c\in\mathbf{s}} w_c^{m_c}\right) e_k(\mathbf{m}\mathbf{x_s})\\
&= f_i^{\mathbf{0}}(x) + (n-k)!k! \sum_{\substack{\mathbf{m}\geq\mathbf{0}_{|\mathbf{s}|}\\ |\mathbf{m}|=n}} \left(\prod_{c\in\mathbf{s}} \frac{w_c^{m_c}}{m_c!}\right) \sum_{\substack{\mathbf{q}\geq\mathbf{0}_{|\mathbf{s}|}\\ \mathbf{q}\leq\mathbf{m}\\ |\mathbf{q}|=k}} \prod_{c\in\mathbf{s}} x_c^{q_c}\binom{m_c}{q_c}\\
&= f_i^{\mathbf{0}}(x) + (n-k)!k! \sum_{\substack{\mathbf{m}\geq\mathbf{0}_{|\mathbf{s}|}\\ |\mathbf{m}|=n}} \sum_{\substack{\mathbf{q}\geq\mathbf{0}_{|\mathbf{s}|}\\ \mathbf{q}\leq\mathbf{m}\\ |\mathbf{q}|=k}} \prod_{c\in\mathbf{s}} \frac{w_c^{m_c-q_c}(w_c x_c)^{q_c}}{(m_c-q_c)!q_c!}.
\end{aligned}$$

Define $p_c := m_c - q_c$. Then, we can write this in terms of $\mathbf{p}$ *as*

$$\begin{aligned}\hat{f}_i(x;\mathbf{w_s},\mathbf{x_s}) &= f_i^{\mathbf{0}}(x) + (n-k)!k! \sum_{\substack{\mathbf{p}\geq\mathbf{0}_{|\mathbf{s}|}\\ \mathbf{q}\geq\mathbf{0}_{|\mathbf{s}|}\\ |\mathbf{p}|=n-k\\ |\mathbf{q}|=k}} \prod_{c\in\mathbf{s}} \frac{w_c^{p_c}(w_c x_c)^{q_c}}{p_c!q_c!}\\
&= f_i^{\mathbf{0}}(x) + \left[(n-k)! \sum_{\substack{\mathbf{p}\geq\mathbf{0}_{|\mathbf{s}|}\\ |\mathbf{p}|=n-k}} \prod_{c\in\mathbf{s}} \frac{w_c^{p_c}}{p_c!}\right]\left[k! \sum_{\substack{\mathbf{q}\geq\mathbf{0}_{|\mathbf{s}|}\\ |\mathbf{q}|=k}} \prod_{c\in\mathbf{s}} \frac{(w_c x_c)^{q_c}}{q_c!}\right]\\
&= f_i^{\mathbf{0}}(x) + \left(\sum_{c\in\mathbf{s}} w_c\right)^{n-k}\left(\sum_{c\in\mathbf{s}} w_c x_c\right)^{k},\end{aligned}$$

which completes our proof. □

For completeness sake, we present in Theorem 5.3.24 the inverse result of Theorem 5.3.16. That is, we express the basis components in terms of the oracle components. In this result, we use a generalization of Stirling numbers called r-Stirling numbers, which are defined in Broder (1984).

Definition 5.3.22. *The unsigned r-Stirling numbers of the first kind, $\mathcal{S}_1^r(n,k)$, with $r,n,k \geq 0$, are given by the recurrence relation*

$$\mathcal{S}_1^r(n,k) = (n-1)\mathcal{S}_1^r(n-1,k) + \mathcal{S}_1^r(n-1,k-1), \quad n > r, k > 0,$$

together with the boundary conditions

$$\begin{aligned}\mathcal{S}_1^r(r,k) &= \delta_{r,k},\\ \mathcal{S}_1^r(n,k) &= 0 \quad n < r,\\ \mathcal{S}_1^r(n,0) &= 0 \quad n > r.\end{aligned}$$

□

Remark 5.3.23. *Note that $\mathcal{S}_1^0(n,k) = \mathcal{S}_1(n,k)$. Moreover, $\mathcal{S}_1^1(n,k) = \mathcal{S}_1(n,k)$ when $n > 0$.* □

We denote by $\hat{\mathcal{F}}_i^{\leq\mathbf{K}}$ the subset of all $\hat{f}_i \in \hat{\mathcal{F}}_i$ such that all of their non-zero coupling components are inside the subset $\{f_i^{\mathbf{k}}\}_{\mathbf{k}\leq\mathbf{K}}$. From Lemma 5.2.9, this forms a subspace.

Theorem 5.3.24. *For every finite order $\hat{f}_i \in \hat{\mathcal{F}}_i^{<\infty}$, we can express the set of basis components $\{{}^b f_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}}$, which have the properties as in Definition 5.3.3, in terms of its oracle components $\hat{f}_i$. In particular, for any $\mathbf{K} \geq \mathbf{0}_{|T|}$ such that $\hat{f}_i \in \hat{\mathcal{F}}_i^{\leq\mathbf{K}}$, we have that*

$${}^b f_i^{\mathcal{K}(\mathbf{s})}(x;\mathbf{w_s},\mathbf{x_s}) = (-1)^{|\mathbf{s}|} \sum_{\bar{\mathbf{s}}\subseteq\mathbf{s}} \sum_{\substack{\mathbf{M}\geq\mathbf{1}_{|\bar{\mathbf{s}}|}\\ \mathcal{K}(\mathbf{M}\bar{\mathbf{s}})\leq\mathbf{K}}} \frac{(-1)^{|\mathbf{M}|}}{\prod_{c\in\bar{\mathbf{s}}} M_c} \left[\prod_{j\in T} C(K_j, |\mathbf{M}_j|, |\mathbf{s}_j \setminus \bar{\mathbf{s}}_j|)\right] \hat{f}_i(x;\mathbf{M},\mathbf{w}_{\bar{\mathbf{s}}},\mathbf{x}_{\bar{\mathbf{s}}}) \quad (5.32)$$

where $C(K,M,r)$*, with* $K \geq M \geq 0$ *and* $r \geq 0$*, is defined as*

$$C(K,M,r) := \frac{r!}{(K-M)!}\mathcal{S}_1^{M+1}(K+1, r+M+1). \tag{5.33}$$

□

In order to prove this, we require Lemma 5.3.25, which is proved in Appendix A.2.

Lemma 5.3.25. *For* $\mathbf{M} \geq \mathbf{1}_k$*, with* $n \geq |\mathbf{M}|$ *and* $k, r \geq 0$*, we have that*

$$\sum_{\substack{\mathbf{m}\geq\mathbf{M}\\ \mathbf{p}\geq\mathbf{1}_r\\ |\mathbf{m}|+|\mathbf{p}|\leq n}} \prod_{i=1}^{k}\binom{m_i-1}{M_i-1}\prod_{j=1}^{r}\frac{1}{p_j} = \frac{r!}{(n-|\mathbf{M}|)!}\mathcal{S}_1^{|\mathbf{M}|+1}(n+1, r+|\mathbf{M}|+1). \tag{5.34}$$

□

Proof of Theorem 5.3.24. We first note that Equation (5.4) can be generalized into

$$f_i^{\mathcal{K}(\mathbf{ms})}(x; \mathbf{m}, \mathbf{w_s}, \mathbf{x_s}) = \sum_{\substack{\overline{\mathbf{m}}\geq\mathbf{0}_{|\mathbf{s}|}\\ \overline{\mathbf{m}}\leq\mathbf{m}}} \left[\prod_{c\in\mathbf{s}}\binom{m_c}{\overline{m}_c}\right](-1)^{|\mathbf{m}|-|\overline{\mathbf{m}}|}\hat{f}_i(x; \overline{\mathbf{m}}, \mathbf{w_s}, \mathbf{x_s}).$$

Plugging in this result into Equation (5.19), we obtain

$$\begin{aligned}
{}^{b}f_i^{\mathcal{K}(\mathbf{s})}(x; \mathbf{w_s}, \mathbf{x_s}) &= \sum_{\substack{\mathbf{m}\geq\mathbf{1}_{|\mathbf{s}|}\\ \mathcal{K}(\mathbf{ms})\leq\mathbf{K}}} \frac{(-1)^{|\mathbf{m}|-|\mathbf{s}|}}{\prod_{c\in\mathbf{s}} m_c} \sum_{\substack{\overline{\mathbf{m}}\geq\mathbf{0}_{|\mathbf{s}|}\\ \overline{\mathbf{m}}\leq\mathbf{m}}} \left[\prod_{c\in\mathbf{s}}\binom{m_c}{\overline{m}_c}\right](-1)^{|\mathbf{m}|-|\overline{\mathbf{m}}|}\hat{f}_i(x; \overline{\mathbf{m}}, \mathbf{w_s}, \mathbf{x_s})\\
&= (-1)^{|\mathbf{s}|}\sum_{\substack{\mathbf{m}\geq\mathbf{1}_{|\mathbf{s}|}\\ \mathcal{K}(\mathbf{ms})\leq\mathbf{K}}}\sum_{\substack{\overline{\mathbf{m}}\geq\mathbf{0}_{|\mathbf{s}|}\\ \overline{\mathbf{m}}\leq\mathbf{m}}} \left[\prod_{c\in\mathbf{s}}\frac{\binom{m_c}{\overline{m}_c}}{m_c}\right](-1)^{|\overline{\mathbf{m}}|}\hat{f}_i(x; \overline{\mathbf{m}}, \mathbf{w_s}, \mathbf{x_s}).
\end{aligned}$$

Rearranging the two sums we get

$${}^{b}f_i^{\mathcal{K}(\mathbf{s})}(x; \mathbf{w_s}, \mathbf{x_s}) = (-1)^{|\mathbf{s}|}\sum_{\substack{\overline{\mathbf{m}}\geq\mathbf{0}_{|\mathbf{s}|}\\ \mathcal{K}(\overline{\mathbf{m}}\mathbf{s})\leq\mathbf{K}}}(-1)^{|\overline{\mathbf{m}}|}\left[\sum_{\substack{\mathbf{m}\geq\overline{\mathbf{m}}\\ \mathbf{m}\geq\mathbf{1}_{|\mathbf{s}|}\\ \mathcal{K}(\mathbf{ms})\leq\mathbf{K}}}\prod_{c\in\mathbf{s}}\frac{\binom{m_c}{\overline{m}_c}}{m_c}\right]\hat{f}_i(x; \overline{\mathbf{m}}, \mathbf{w_s}, \mathbf{x_s}).$$

We now break the first sum into

$$\sum_{\substack{\overline{\mathbf{m}}\geq\mathbf{0}_{|\mathbf{s}|}\\ \mathcal{K}(\overline{\mathbf{m}}\mathbf{s})\leq\mathbf{K}}} = \sum_{\overline{\mathbf{s}}\subseteq\mathbf{s}}\sum_{\substack{\mathbf{M}\geq\mathbf{1}_{|\overline{\mathbf{s}}|}\\ \mathcal{K}(\mathbf{M}\overline{\mathbf{s}})\leq\mathbf{K}}},$$

that is, we correspond a given $\overline{\mathbf{m}} \geq \mathbf{0}_{|\mathbf{s}|}$ to the subset $\overline{\mathbf{s}} \subseteq \mathbf{s}$, of its non-zero entries. Therefore, we have that $\overline{\mathbf{m}}_{\overline{\mathbf{s}}} = \mathbf{M}$ and $\overline{\mathbf{m}}_{\mathbf{s}\setminus\overline{\mathbf{s}}} = \mathbf{0}_{|\mathbf{s}\setminus\overline{\mathbf{s}}|}$. Note that according to this split, the condition $\mathbf{m} \geq \overline{\mathbf{m}}$ becomes $\mathbf{m}_{\overline{\mathbf{s}}} \geq \mathbf{M}$ and $\mathbf{m}_{\mathbf{s}\setminus\overline{\mathbf{s}}} \geq \mathbf{0}_{|\mathbf{s}\setminus\overline{\mathbf{s}}|}$. On the other hand, the condition $\mathbf{m} \geq \mathbf{1}_{|\mathbf{s}|}$ becomes $\mathbf{m}_{\overline{\mathbf{s}}} \geq \mathbf{1}_{|\overline{\mathbf{s}}|}$

and $\mathbf{m}_{\mathbf{s}\setminus\bar{\mathbf{s}}} \geq \mathbf{1}_{|\mathbf{s}\setminus\bar{\mathbf{s}}|}$. Out of the resulting four conditions, the non-redundant ones are clearly $\mathbf{m}_{\bar{\mathbf{s}}} \geq \mathbf{M}$ and $\mathbf{m}_{\mathbf{s}\setminus\bar{\mathbf{s}}} \geq \mathbf{1}_{|\mathbf{s}\setminus\bar{\mathbf{s}}|}$. This results in

$$
\begin{aligned}
&{}^{b}f_i^{\mathcal{K}(\mathbf{s})}(x;\mathbf{w_s},\mathbf{x_s})\\
&= (-1)^{|\mathbf{s}|}\sum_{\bar{\mathbf{s}}\subseteq\mathbf{s}}\sum_{\substack{\mathbf{M}\geq\mathbf{1}_{|\bar{\mathbf{s}}|}\\ \mathcal{K}(\mathbf{M}\bar{\mathbf{s}})\leq\mathbf{K}}}(-1)^{|\mathbf{M}|}\left[\sum_{\substack{\mathbf{m}_{\bar{\mathbf{s}}}\geq\mathbf{M}\\ \mathbf{m}_{\mathbf{s}\setminus\bar{\mathbf{s}}}\geq\mathbf{1}_{|\mathbf{s}\setminus\bar{\mathbf{s}}|}\\ \mathcal{K}(\mathbf{ms})\leq\mathbf{K}}}\prod_{c\in\bar{\mathbf{s}}}\frac{\binom{m_c}{M_c}}{m_c}\prod_{d\in\mathbf{s}\setminus\bar{\mathbf{s}}}\frac{1}{m_d}\right]\hat{f}_i(x;\mathbf{M},\mathbf{w}_{\bar{\mathbf{s}}},\mathbf{x}_{\bar{\mathbf{s}}})\\
&= (-1)^{|\mathbf{s}|}\sum_{\bar{\mathbf{s}}\subseteq\mathbf{s}}\sum_{\substack{\mathbf{M}\geq\mathbf{1}_{|\bar{\mathbf{s}}|}\\ \mathcal{K}(\mathbf{M}\bar{\mathbf{s}})\leq\mathbf{K}}}\frac{(-1)^{|\mathbf{M}|}}{\prod_{c\in\bar{\mathbf{s}}}M_c}\left[\sum_{\substack{\mathbf{m}_{\bar{\mathbf{s}}}\geq\mathbf{M}\\ \mathbf{m}_{\mathbf{s}\setminus\bar{\mathbf{s}}}\geq\mathbf{1}_{|\mathbf{s}\setminus\bar{\mathbf{s}}|}\\ \mathcal{K}(\mathbf{ms})\leq\mathbf{K}}}\prod_{c\in\bar{\mathbf{s}}}\binom{m_c-1}{M_c-1}\prod_{d\in\mathbf{s}\setminus\bar{\mathbf{s}}}\frac{1}{m_d}\right]\hat{f}_i(x;\mathbf{M},\mathbf{w}_{\bar{\mathbf{s}}},\mathbf{x}_{\bar{\mathbf{s}}})\,.
\end{aligned}
$$

Note that $\frac{\binom{m_c}{M_c}}{m_c} = \frac{\binom{m_c-1}{M_c-1}}{M_c}$, whenever $M_c \geq 1$.
We now show that the expression in brackets gives us $\prod_{j\in T} C(K_j,|\mathbf{M}_j|,|\mathbf{s}_j\setminus\bar{\mathbf{s}}_j|)$, as given by Equation (5.33). We rearrange it by breaking all the multi-indices according to the typing of their cells. That is,

$$
\prod_{j\in T}\sum_{\substack{\mathbf{m}_{\bar{\mathbf{s}}_j}\geq\mathbf{M}_j\\ \mathbf{m}_{\mathbf{s}_j\setminus\bar{\mathbf{s}}_j}\geq\mathbf{1}_{|\mathbf{s}_j\setminus\bar{\mathbf{s}}_j|}\\ |\mathbf{m}_{\bar{\mathbf{s}}_j}|+|\mathbf{m}_{\mathbf{s}_j\setminus\bar{\mathbf{s}}_j}|\leq K_j}}\prod_{c\in\bar{\mathbf{s}}_j}\binom{m_c-1}{M_c-1}\prod_{d\in\mathbf{s}_j\setminus\bar{\mathbf{s}}_j}\frac{1}{m_d},
$$

and from Lemma 5.3.25, the result is proven. ■

Remark 5.3.26. *Note that if* $\hat{f}_i \in \hat{\mathcal{F}}_i^{\leq\mathbf{K}_1}\cap\hat{\mathcal{F}}_i^{\leq\mathbf{K}_2}$*, with* $\mathbf{K}_1 \neq \mathbf{K}_2$*, the associated coefficients* $C(K_j,|\mathbf{M}_j|,|\mathbf{s}_j\setminus\bar{\mathbf{s}}_j|)$ *will be different when we apply Equation* (5.32) *to* $\mathbf{K}_1$ *and* $\mathbf{K}_2$*.*
The fact that there are multiple valid formulas that express $\{{}^{b}f_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}}$ *as a function of* $\hat{f}_i$ *might seem unexpected. One way to convince ourselves that this is reasonable, is to consider a simple case like* $\hat{f}_i \in \hat{\mathcal{F}}_i^{\leq\mathbf{0}}$*. In this case, we have that* ${}^{b}f_i^{\mathbf{0}}(x) = \hat{f}_i(x;\mathbf{w_s},\mathbf{x_s})$ *and it is easy to get creative and find multiple formulas that are all valid under the (very strict) assumption that* $\hat{f}_i \in \hat{\mathcal{F}}_i^{\leq\mathbf{0}}$*.* □

Remark 5.3.27. *Note that the coefficients* $C(K_j,|\mathbf{M}_j|,|\mathbf{s}_j\setminus\bar{\mathbf{s}}_j|)$ *diverge as* $\mathbf{K}\to\infty$*. In particular,* $C(K_j,1,0) = \mathcal{S}_1^2(K_j+1,2)/(K_j-1)! = K_j!/(K_j-1)! = K_j$ *for* $K_j \geq 1$*. Therefore, the limit case of Theorem 5.3.24 does not give us a universal formula that works for all* $\hat{f}_i \in \hat{\mathcal{F}}_i^{<\infty}$*.* □

5.3.4 Infinite coupling order

All the results of Section 5.3.3 fall under the assumption that the oracle functions are in $\hat{\mathcal{F}}_i^{<\infty}$. That is, they have finite order. There are, however, plentiful useful functions that lie outside this subspace, such as the exponential function in Example 5.2.22. Our goal is to create a useful extension of this theory that applies to at least some important functions, such as the exponential and the trigonometric functions. The first idea that comes to mind is to simply allow the family of basis components $\{{}^b f_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}}$ to have infinite support. There is an important issue with such an approach. When dealing with infinite sums we are actually talking about limits on a sequence of partial sums. For this to be well-defined we need to be clear about the meaning of infinite sums of the type $\sum_{\mathbf{m}\geq\mathbf{1}_{|\mathbf{s}|}} a_{\mathbf{m}}$. There are plenty of possible definitions, with some of the more obvious ones being

$$\lim_{N\to\infty}\sum_{n=0}^{N}\sum_{\substack{\mathbf{m}\geq\mathbf{1}_{|\mathbf{s}|}\\ |\mathbf{m}|=n}} a_{\mathbf{m}}, \quad \text{or} \quad \lim_{N\to\infty}\sum_{n=0}^{N}\sum_{\substack{\mathbf{m}\geq\mathbf{1}_{|\mathbf{s}|}\\ \max(\mathbf{m})=n}} a_{\mathbf{m}}.$$

However, there is no clear reason for why one definition would be preferable to the other. If we chose one of them and developed our theory based on that, we would only be restricting ourselves to that choice. A different (and better) approach is to simply choose to give up on a $\{{}^b f_i^{\mathbf{k}}\}_{\mathbf{k}\geq\mathbf{0}_{|T|}}$ representation for oracle components outside $\hat{\mathcal{F}}_i^{<\infty}$ and use the following results instead.

Lemma 5.3.28. *Consider $\hat{\mathcal{F}}_i$ and $\hat{\mathcal{F}}_T$ such that the related sets $\{\mathbb{Y}_j\}_{j\in T}$ are Hausdorff spaces. Then, $\hat{\mathcal{F}}_i$ and $\hat{\mathcal{F}}_T$ are sequentially closed in the topology of pointwise convergence (product topology).* □

Proof. Consider a sequence of functions $({}^N\hat{f}_i)_{N\in\mathbb{N}}$, with ${}^N\hat{f}_i \in \hat{\mathcal{F}}_i$ for all $N\in\mathbb{N}$, such that it converges pointwise to some function $\hat{f}_i$. That is,

$$\lim_{N\to\infty} {}^N\hat{f}_i(x;\mathbf{w},\mathbf{x}) = \hat{f}_i(x;\mathbf{w},\mathbf{x}),$$

for all $x\in\mathbb{X}_i$, $\mathbf{x}\in\mathbb{X}^{\mathbf{k}}$, $\mathbf{w}\in\mathcal{M}_i^{\mathbf{k}}$, for any given $\mathbf{k}\geq\mathbf{0}_{|T|}$. Given a permutation matrix σ of appropriate dimension, then

$$\lim_{N\to\infty} {}^N\hat{f}_i(x;\sigma\mathbf{w},\sigma\mathbf{x}) = \hat{f}_i(x;\sigma\mathbf{w},\sigma\mathbf{x}).$$

Note that from assumption, Equation (2.2) is satisfied for every ${}^N\hat{f}_i$. Therefore, these two sequences are the same. Since $\mathbb{Y}_i$ is Hausdorff, we know that the limit of a convergent sequence is unique, which implies $\hat{f}_i(x;\mathbf{w},\mathbf{x}) = \hat{f}_i(x;\sigma\mathbf{w},\sigma\mathbf{x})$. That is, $\hat{f}_i$ also satisfies Equation (2.2). The same reasoning applies with respect to Equations (2.3) and (2.4).
Therefore, $\hat{f}_i\in\hat{\mathcal{F}}_i$, which means that $\hat{\mathcal{F}}_i$ is sequentially closed. Since the product of sequentially closed sets is sequentially closed, $\hat{\mathcal{F}}_T$ is also sequentially closed. ■

Corollary 5.3.29. *Consider the related set $\mathbb{Y}_i$ to be a Hausdorff vector space. Then, for every sequence $({}^N\hat{f}_i)_{N\in\mathbb{N}}$, with ${}^N\hat{f}_i\in\hat{\mathcal{F}}_i^{<\infty}$ for all $N\in\mathbb{N}$ such that $\hat{f}_i := \lim_{N\to\infty} {}^N\hat{f}_i$ converges pointwise, we have that $\hat{f}_i\in\hat{\mathcal{F}}_i$.* □

Proof. This is direct from Lemma 5.3.28 and $\hat{\mathcal{F}}_i^{<\infty}\subset\hat{\mathcal{F}}_i$. We assume $\mathbb{Y}_i$ is a vector space so that $\hat{\mathcal{F}}_i^{<\infty}$ can be defined. ■

This provides us with a framework that allows us to build a set of oracle components with infinite order, in particular the ones in $scl(\hat{\mathcal{F}}_i^{<\infty})$, the sequential closure of $\hat{\mathcal{F}}_i^{<\infty}$. We illustrate this with the following example.

Example 5.3.30. *Consider the sequence of oracle components with finite coupling order $({}^N\hat{f}_i)_{N\in\mathbb{N}}$ such that the basis components of ${}^N\hat{f}_i \in \hat{\mathcal{F}}_i^{<\infty}$ are given according to*

$$^{bN}f_i^{|\mathbf{s}|}(x;\mathbf{w_s},\mathbf{x_s}) = \begin{cases} a_{|\mathbf{s}|}|\mathbf{s}|!\prod_{c\in\mathbf{s}}(w_c x_c) & 0<|\mathbf{s}|\leq N, \\ f_i^0(x) & |\mathbf{s}|=0, \\ 0 & \textit{otherwise}. \end{cases}$$

From Example 5.3.19, we know that this corresponds to the oracle component

$$^N\hat{f}_i(x;\mathbf{w_s},\mathbf{x_s}) = f_i^0(x) + \sum_{n=1}^{N} a_n \left(\sum_{c\in\mathbf{s}} w_c x_c\right)^n.$$

Then, if the infinite series given by $F(x) = \sum_{n=1}^{\infty} a_n x^n$ converges for all x, we know from Corollary 5.3.29 that $\hat{f}_i := \lim_{N\to\infty} {}^N\hat{f}_i$ is given by

$$\hat{f}_i(x;\mathbf{w_s},\mathbf{x_s}) = f_i^0(x) + F\left(\sum_{c\in\mathbf{s}} w_c x_c\right)$$

and it is a valid oracle component. □

Remark 5.3.31. *Note that this example covers functions such as $F(x)=\exp(x)-1$, $F(x)=\sin(x)$ and $F(x)=\cos(x)-1$.* □

Lemma 5.3.32. *The set $scl(\hat{\mathcal{F}}_i^{<\infty})$ is a vector space.* □

Proof. Consider $\hat{f}_i,\hat{g}_i \in scl(\hat{\mathcal{F}}_i^{<\infty})$. From assumption, there are sequences $({}^N\hat{f}_i)_{N\in\mathbb{N}}$, $({}^N\hat{g}_i)_{N\in\mathbb{N}}$ with ${}^N\hat{f}_i, {}^N\hat{g}_i \in \hat{\mathcal{F}}_i^{<\infty}$ such that $\lim_{N\to\infty} {}^N\hat{f}_i = \hat{f}_i$ and $\lim_{N\to\infty} {}^N\hat{g}_i = \hat{g}_i$. Then, the elements of the sequence $(\alpha^N\hat{f}_i + {}^N\hat{g}_i)_{N\in\mathbb{N}}$ are also in $\hat{\mathcal{F}}_i^{<\infty}$ and the sequence converges into $\alpha\hat{f}_i+\hat{g}_i$. Therefore $\alpha\hat{f}_i+\hat{g}_i \in scl(\hat{\mathcal{F}}_i^{<\infty})$ and $scl(\hat{\mathcal{F}}_i^{<\infty})$ is a vector space. ■

Remark 5.3.33. *Note that $\hat{\mathcal{F}}_i^{<\infty} \subseteq scl(\hat{\mathcal{F}}_i^{<\infty}) \subseteq \hat{\mathcal{F}}_i$.* □

5.4 Extension for exogenous inputs and inner cell parameters

Using the extension described in Section 2.5, we can similarly extend the coupling components $f_i^{\mathbf{k}}$ and basis components ${}^b f_i^{\mathbf{k}}$ to be defined on

$$f_i^{\mathbf{k}}, {}^b f_i^{\mathbf{k}} \colon \mathbb{X}_i \times \mathbb{P}_i \times \mathbb{U}_i \times \mathcal{M}_i^{\mathbf{k}} \times \mathbb{X}^{\mathbf{k}} \to \mathbb{Y}_i. \tag{5.35}$$

It is clear that the decomposition schemes and related results described in Sections 5.2 and 5.3 apply to the extended framework with essentially the same formulas. In the following example we show how to construct a valid oracle component using the naturally extended version of Theorem 5.3.16.

Example 5.4.1. *Consider a single-type finite order oracle component $\hat{f}_i \in \hat{\mathcal{F}}_i^{<\infty}$ with basis components $\{{}^b f_i^k\}_{k\geq 0}$ such that, for some fixed $n > 0$*

$$
{}^b f_i^{|\mathbf{s}|}(x,p,u;\mathbf{w_s},\mathbf{x_s}) = \begin{cases} u^2 n! \prod_{c\in\mathbf{s}}(w_c x_c) & |\mathbf{s}| = n, \\ f_i^0(x,p,u) & |\mathbf{s}| = 0, \\ 0 & \textit{otherwise}. \end{cases}
$$

It is clear that $\{{}^b f_i^k\}_{k\geq 0}$ satisfy items 1 and 2 and Corollary 5.3.4 of Definition 5.3.3. Using the direct extension of Theorem 5.3.16 to this framework, we can find the corresponding oracle component directly. That is,

$$
\begin{aligned}
\hat{f}_i(x,p,u;\mathbf{w_s},\mathbf{x_s}) &= \sum_{\mathbf{m}\geq\mathbf{0}_{|\mathbf{s}|}} \frac{1}{\prod_{c\in\mathbf{s}} m_c!}\,{}^b f_i^{|\mathbf{m}|}(x,p,u;\mathbf{m},\mathbf{w_s},\mathbf{x_s}) \\
&= f_i^0(x,p,u) + u^2 n! \sum_{\substack{\mathbf{m}\geq\mathbf{0}_{|\mathbf{s}|}\\ |\mathbf{m}|=n}} \prod_{c\in\mathbf{s}} \frac{(w_c x_c)^{m_c}}{m_c!} \\
&= f_i^0(x,p,u) + u^2 \left(\sum_{c\in\mathbf{s}} w_c x_c\right)^n .
\end{aligned}
$$

□

Chapter 6

Conclusion

This thesis generalizes the theory of coupled cell networks to the general weighted case.
We found it simpler to encode the notion of in-neighborhood equivalence through items 1 to 3 of Definition 2.4.1, which act as generators of a set of equality constraints, instead of defining them using a pullback map on a groupoid of bijections, as in the original CCN formalism.
In this work, we do not assume, in geral, any prior structure on the cell domains $\mathbb{X}_i$ and codomains $\mathbb{Y}_i$ or any smoothness requirement in order for a function to be admissible. While this is quite usual and reasonable for continuous-time systems, it is not so for discrete-time systems. Such assumptions are application-dependent and therefore, should not be built in the general definitions. One can always particularize by adding such constraints at a later point. Furthermore, we do not require the networks to obey the "consistency condition", which would force the different monoids $\{\mathcal{M}_{ij}\}_{i,j\in T}$ to be pairwise disjoint. We only operate ($\|$) and compare elements within the same monoid $\mathcal{M}_{ij}$, therefore, such condition is unnecessary.
The most important contribution of this work is the definition of oracle components. The crucial difference with respect to the previous formalism is that the notion of in-neighborhood equivalence is imposed on all (finite) in-neighborhoods of a cell, and not just the ones that might appear in our particular network of interest, according to its size. In this sense, it is more constrained that in the original formalism, however, we argue that this is a very reasonable constraint. In particular, note that from a physical point of view, the existence of function that describes how a cell behaves under arbitrary (finite) in-neighborhoods is not too much to ask for. No matter how large, complicated, or even nonsensical is the way that we connect a physical system, there will always be an equation that describes its evolution, regardless of how dramatic the result is (e.g., the whole system crashing or burning). That is, the underlying oracle components always "exists".
Furthermore, note that the idea of specifying how a cell behaves under arbitrary in-neighborhoods is not new. This is something that in practice is done everywhere, through specific expressions according to the particular application. The contribution here was to formalize the idea underlying such expressions as a mathematical object in its own right and generalizing it by only requiring it to satisfy the notions of in-neighborhood equivalence.
Note that this approach has the very important advantage that given an oracle function, the ad-

missible function associated with every network (with finite neighborhoods) is automatically (and uniquely) defined. On the other hand, as we have shown in Example 2.4.7, by constructing admissible functions while only enforcing cross-compatibility between the particular types of in-neighborhoods that we are interested in, does not gives us any guarantees that such functions could then be further extended for other types of in-neighborhoods. In such a case, we would have assigned an admissible function to a network while claiming that some other networks living in the same universe and evolving according to the same laws would simply not have an admissible function that specifies its dynamical evolution. Such a scenario would be non-physical and the way of guaranteeing that such a situation can never arise is through the proposed approach.

Another argument in favor of the oracle component is the fact that having the behavior of a cell completely specified for any conceivable neighborhood made it not only possible but also very natural to ask the right questions which led to the development of the decompositions in Sections 5.2 and 5.3. We consider these decompositions to be the part of this work that has the most potential for practical application.

Due to the rise in interest in admissible functions with higher order, non-pairwise structure, other works in the literature have taken a different approach and focused in the construction of these terms by making the network structure more complicated (e.g., hypernetworks and simplicial complexes). We would suggest a similar approach to the one present in this work to be applied to the study of these more general structures. In particular, to start from the absolutely minimal assumptions (e.g., in our case it was the notion of in-neighborhood equivalence) and then reverse engineer the degrees of freedom from the ground up. This should be done in a systematic manner in order to ensure that there are no missing terms from the stated formulas. For this reason, we argue that extensions of the concept of oracle function for higher-order network structures should be investigated.

References

Manuela Aguiar and Ana Dias. Synchrony and antisynchrony in weighted networks. *SIAM Journal on Applied Dynamical Systems*, 20(3):1382–1420, 2021.

Manuela Aguiar, Christian Bick, and Ana Dias. Network dynamics with higher-order interactions: Coupled cell hypernetworks for identical cells and synchrony. *arXiv preprint arXiv:2201.09379*, 2022.

Manuela AD Aguiar and Ana Paula S Dias. Synchronization and equitable partitions in weighted networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 28(7):073105, 2018.

Manuela AD Aguiar, Ana Paula S Dias, and Flora Ferreira. Patterns of synchrony for feed-forward and auto-regulation feed-forward neural networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 27(1):013103, 2017.

John W Aldis. A polynomial time algorithm to determine maximal balanced equivalence relations. *International Journal of Bifurcation and Chaos*, 18(02):407–427, 2008.

Alex Arenas, Albert Díaz-Guilera, Jurgen Kurths, Yamir Moreno, and Changsong Zhou. Synchronization in complex networks. *Physics reports*, 469(3):93–153, 2008.

Peter Ashwin and Ana Rodrigues. Hopf normal form with sn symmetry and reduction to systems of nonlinearly coupled phase oscillators. *Physica D: Nonlinear Phenomena*, 325:14–24, 2016.

Jørgen Bang-Jensen and Gregory Z Gutin. *Digraphs: theory, algorithms and applications*. Springer Science & Business Media, 2008.

Federico Battiston, Giulia Cencetti, Iacopo Iacopini, Vito Latora, Maxime Lucas, Alice Patania, Jean-Gabriel Young, and Giovanni Petri. Networks beyond pairwise interactions: structure and dynamics. *Physics Reports*, 874:1–92, 2020.

Federico Battiston, Enrico Amico, Alain Barrat, Ginestra Bianconi, Guilherme Ferraz de Arruda, Benedetta Franceschiello, Iacopo Iacopini, Sonia Kéfi, Vito Latora, Yamir Moreno, et al. The physics of higher-order interactions in complex systems. *Nature Physics*, 17(10):1093–1098, 2021.

Christian Bick, Peter Ashwin, and Ana Rodrigues. Chaos in generically coupled phase oscillator networks with nonpairwise interactions. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 26(9):094814, 2016.

Christian Bick, Elizabeth Gross, Heather A Harrington, and Michael T Schaub. What are higher-order networks? *arXiv preprint arXiv:2104.11329*, 2021.

Andrei Z Broder. The r-stirling numbers. *Discrete Mathematics*, 49(3):241–259, 1984.

Louis Comtet. *Advanced Combinatorics: The art of finite and infinite expansions*. Springer Science & Business Media, 1974.

B. A. Davey and H. A. Priestley. *Introduction to Lattices and Order*. Cambridge University Press, 2 edition, 2002.

Florian Dörfler and Francesco Bullo. Synchronization in complex networks of phase oscillators: A survey. *Automatica*, 50(6):1539–1564, 2014.

Martin Golubitsky and Ian Stewart. Nonlinear dynamics of networks: the groupoid formalism. *Bulletin of the american mathematical society*, 43(3):305–364, 2006.

Martin Golubitsky, Ian Stewart, and Andrei Török. Patterns of synchrony in coupled cell networks with multiple arrows. *SIAM Journal on Applied Dynamical Systems*, 4(1):78–100, 2005.

F Kuo, I Sloan, Grzegorz Wasilkowski, and Henryk Woźniakowski. On decompositions of multivariate functions. *Mathematics of computation*, 79(270):953–966, 2010.

Raoul-Martin Memmesheimer and Marc Timme. Non-additive coupling enables propagation of synchronous spiking activity in purely random networks. *PLoS computational biology*, 8(4): e1002384, 2012.

John M Neuberger, Nandor Sieben, and James W Swift. Invariant synchrony subspaces of sets of matrices. *SIAM Journal on Applied Dynamical Systems*, 19(2):964–993, 2020.

Eddie Nijholt and Lee DeVille. Dynamical systems defined on simplicial complexes: symmetries, conjugacies, and invariant subspaces. *arXiv preprint arXiv:2204.08350*, 2022.

Francisco A Rodrigues, Thomas K DM Peron, Peng Ji, and Jürgen Kurths. The kuramoto model in complex networks. *Physics Reports*, 610:1–98, 2016.

Pedro M. Sequeira, A. Pedro Aguiar, and João Hespanha. Commutative monoid formalism for weighted coupled cell networks and invariant synchrony patterns. *SIAM Journal on Applied Dynamical Systems*, 20(3):1485–1513, 2021.

Pedro M. Sequeira, João P. Hespanha, and A. Pedro Aguiar. Decomposition of admissible functions in weighted coupled cell networks. *SIAM Journal on Applied Dynamical Systems*, 22(2): 1114–1152, 2023a.

Pedro M. Sequeira, João P. Hespanha, and A. Pedro Aguiar. An in-reachability based classification of invariant synchrony patterns in weighted coupled cell networks. *arXiv preprint arXiv:2306.00725*, 2023b.

Ian Stewart. The lattice of balanced equivalence relations of a coupled cell network. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 143, pages 165–183. Cambridge University Press, 2007.

Ian Stewart, Martin Golubitsky, and Marcus Pivato. Symmetry groupoids and patterns of synchrony in coupled cell networks. *SIAM Journal on Applied Dynamical Systems*, 2(4):609–646, 2003.

Steven H Strogatz and Ian Stewart. Coupled oscillators and biological synchronization. *Scientific American*, 269(6):102–109, 1993.

Robert Tarjan. Depth-first search and linear graph algorithms. *SIAM journal on computing*, 1(2): 146–160, 1972.

Appendix A

Intermediate results

This appendix presents intermediate results that are required for some proofs in the main text. In particular, in Appendix A.1 we derive Lemmas 5.3.6 to 5.3.12, which are used to prove Theorem 5.3.5. Appendix A.2 derives Lemma 5.3.25 which is used to prove Theorem 5.3.24.

A.1 Intermediate results used in Theorem 5.3.5

Lemma A.1.1. *For every $n \in \mathbb{Z}$, $k \in \mathbb{N}$, we have that*

$$\sum_{\substack{\mathbf{m} \geq \mathbf{1}_k \\ |\mathbf{m}|=n}} \frac{n}{\prod_{i=1}^{k} m_i} = \sum_{\substack{\mathbf{m} \geq \mathbf{1}_k \\ |\mathbf{m}|=n-1}} \frac{n-1}{\prod_{i=1}^{k} m_i} + \sum_{\substack{\mathbf{m} \geq \mathbf{1}_{k-1} \\ |\mathbf{m}|=n-1}} \frac{k}{\prod_{i=1}^{k-1} m_i}. \tag{A.1}$$

□

Proof. Using the fact that $|\mathbf{m}| = n$, we can rewrite the left hand side into

$$\sum_{\substack{\mathbf{m} \geq \mathbf{1}_k \\ |\mathbf{m}|=n}} \frac{n}{\prod_{i=1}^{k} m_i} = \sum_{\substack{\mathbf{m} \geq \mathbf{1}_k \\ |\mathbf{m}|=n}} \sum_{j=1}^{k} \frac{m_j}{\prod_{i=1}^{k} m_i} = \sum_{j=1}^{k} \sum_{\substack{\mathbf{m} \geq \mathbf{1}_k \\ |\mathbf{m}|=n}} \frac{1}{\prod_{\substack{i=1 \\ i \neq j}}^{k} m_i}.$$

Note that $m_j \geq 1$, therefore, the quotients are always well-defined. We perform a change of variables by removing the entry j of the multi-index $\mathbf{m}$. The other conditions in the sum have to be adjusted accordingly, in particular, $|\mathbf{m}| = n$ becomes $|\mathbf{m}| \leq n-1$, that is,

$$\sum_{j=1}^{k} \sum_{\substack{\mathbf{m} \geq \mathbf{1}_k \\ |\mathbf{m}|=n}} \frac{1}{\prod_{\substack{i=1 \\ i \neq j}}^{k} m_i} = \sum_{j=1}^{k} \sum_{\substack{\mathbf{m} \geq \mathbf{1}_{k-1} \\ |\mathbf{m}|\leq n-1}} \frac{1}{\prod_{i=1}^{k-1} m_i} = \sum_{\substack{\mathbf{m} \geq \mathbf{1}_{k-1} \\ |\mathbf{m}|\leq n-1}} \frac{k}{\prod_{i=1}^{k-1} m_i}.$$

In summary, we have proven that

$$\sum_{\substack{\mathbf{m} \geq \mathbf{1}_k \\ |\mathbf{m}|=n}} \frac{n}{\prod_{i=1}^{k} m_i} = \sum_{\substack{\mathbf{m} \geq \mathbf{1}_{k-1} \\ |\mathbf{m}|\leq n-1}} \frac{k}{\prod_{i=1}^{k-1} m_i}.$$

This expression is valid for every $n \in \mathbb{Z}$, therefore, changing variable n into $n-1$, one obtains

$$\sum_{\substack{\mathbf{m}\geq\mathbf{1}_k \\ |\mathbf{m}|=n-1}} \frac{n-1}{\prod_{i=1}^{k} m_i} = \sum_{\substack{\mathbf{m}\geq\mathbf{1}_{k-1} \\ |\mathbf{m}|\leq n-2}} \frac{k}{\prod_{i=1}^{k-1} m_i}.$$

These two equations can be merged in the following way

$$\begin{aligned}
\sum_{\substack{\mathbf{m}\geq\mathbf{1}_k \\ |\mathbf{m}|=n}} \frac{n}{\prod_{i=1}^{k} m_i} &= \sum_{\substack{\mathbf{m}\geq\mathbf{1}_{k-1} \\ |\mathbf{m}|\leq n-1}} \frac{k}{\prod_{i=1}^{k-1} m_i} \\
&= \sum_{\substack{\mathbf{m}\geq\mathbf{1}_{k-1} \\ |\mathbf{m}|\leq n-2}} \frac{k}{\prod_{i=1}^{k-1} m_i} + \sum_{\substack{\mathbf{m}\geq\mathbf{1}_{k-1} \\ |\mathbf{m}|=n-1}} \frac{k}{\prod_{i=1}^{k-1} m_i} \\
&= \sum_{\substack{\mathbf{m}\geq\mathbf{1}_k \\ |\mathbf{m}|=n-1}} \frac{n-1}{\prod_{i=1}^{k} m_i} + \sum_{\substack{\mathbf{m}\geq\mathbf{1}_{k-1} \\ |\mathbf{m}|=n-1}} \frac{k}{\prod_{i=1}^{k-1} m_i},
\end{aligned}$$

which concludes the proof. ■

Lemma A.1.2. *For every $n \in \mathbb{Z}$, $k \in \mathbb{N}$, we have that*

$$\sum_{\substack{\mathbf{m}\geq\mathbf{1}_k \\ |\mathbf{m}|=n}} \frac{n}{\prod_{i=1}^{k} m_i!} = \sum_{\substack{\mathbf{m}\geq\mathbf{1}_k \\ |\mathbf{m}|=n-1}} \frac{k}{\prod_{i=1}^{k} m_i!} + \sum_{\substack{\mathbf{m}\geq\mathbf{1}_{k-1} \\ |\mathbf{m}|=n-1}} \frac{k}{\prod_{i=1}^{k-1} m_i!}. \tag{A.2}$$

□

Proof. Using the fact that $|\mathbf{m}| = n$, we can rewrite the left hand side into

$$\sum_{\substack{\mathbf{m}\geq\mathbf{1}_k \\ |\mathbf{m}|=n}} \frac{n}{\prod_{i=1}^{k} m_i!} = \sum_{\substack{\mathbf{m}\geq\mathbf{1}_k \\ |\mathbf{m}|=n}} \sum_{j=1}^{k} \frac{m_j}{\prod_{i=1}^{k} m_i!} = \sum_{j=1}^{k} \sum_{\substack{\mathbf{m}\geq\mathbf{1}_k \\ |\mathbf{m}|=n}} \frac{1}{(m_j-1)!\prod_{\substack{i=1 \\ i\neq j}}^{k} m_i!}.$$

Note that $m_j \geq 1$, therefore, the quotients are always well-defined. We now split the multi-index $\mathbf{m}$ according to whenever $m_j \geq 2$ or $m_j = 1$, that is,

$$\sum_{j=1}^{k} \left(\sum_{\substack{\mathbf{m}\geq\mathbf{1}_k \\ m_j\geq 2 \\ |\mathbf{m}|=n}} \frac{1}{(m_j-1)!\prod_{\substack{i=1 \\ i\neq j}}^{k} m_i!} + \sum_{\substack{\mathbf{m}\geq\mathbf{1}_k \\ m_j=1 \\ |\mathbf{m}|=n}} \frac{1}{\prod_{\substack{i=1 \\ i\neq j}}^{k} m_i!} \right).$$

On the first inner sum, we perform a change of variables so that $m_j - 1$ becomes m_j. This means that the condition $m_j \geq 2$ becomes $m_j \geq 1$. Therefore, we can compress the conditions $\mathbf{m} \geq \mathbf{1}_k$ and $m_j \geq 2$ in the old coordinates into just $\mathbf{m} \geq \mathbf{1}_k$ in the new ones.

On the second inner sum we perform a change of variables by removing the entry j of the multi-index $\mathbf{m}$. For both sums, the conditions $|\mathbf{m}| = n$ become $|\mathbf{m}| = n-1$. That is,

$$\sum_{\substack{\mathbf{m}\geq\mathbf{1}_k\\|\mathbf{m}|=n}} \frac{n}{\prod_{i=1}^{k} m_i!} = \sum_{j=1}^{k}\left(\sum_{\substack{\mathbf{m}\geq\mathbf{1}_k\\|\mathbf{m}|=n-1}} \frac{1}{\prod_{i=1}^{k} m_i!} + \sum_{\substack{\mathbf{m}\geq\mathbf{1}_{k-1}\\|\mathbf{m}|=n-1}} \frac{1}{\prod_{i=1}^{k-1} m_i!}\right)$$
$$= \sum_{\substack{\mathbf{m}\geq\mathbf{1}_k\\|\mathbf{m}|=n-1}} \frac{k}{\prod_{i=1}^{k} m_i!} + \sum_{\substack{\mathbf{m}\geq\mathbf{1}_{k-1}\\|\mathbf{m}|=n-1}} \frac{k}{\prod_{i=1}^{k-1} m_i!},$$

which concludes the proof. ■

We are now ready to introduce our new formulas for the Stirling numbers of the first and second kinds.

Theorem A.1.3. *The unsigned Stirling numbers of the first kind, $\mathcal{S}_1(n,k)$, with $n,k\geq 0$ are given by*

$$\mathcal{S}_1(n,k) = \frac{n!}{k!}\sum_{\substack{\mathbf{m}\geq\mathbf{1}_k\\|\mathbf{m}|=n}} \frac{1}{\prod_{i=1}^{k} m_i}. \tag{A.3}$$

□

Proof. We have to prove that the right hand side of Equation (A.3) has the properties of Definition 5.3.1.
Consider $n,k=0$, the initial condition is satisfied since the only valid argument of the sum is the 0-tuple. For both $n>0$, $k=0$ and $n=0$, $k>0$ there are no valid arguments in the sum, which results in zero and those initial conditions are also satisfied.
For the remaining values, $n,k>0$, we have to show that they follow the recurrence relation, that is,

$$\frac{n!}{k!}\sum_{\substack{\mathbf{m}\geq\mathbf{1}_k\\|\mathbf{m}|=n}} \frac{1}{\prod_{i=1}^{k} m_i} = (n-1)\frac{(n-1)!}{k!}\sum_{\substack{\mathbf{m}\geq\mathbf{1}_k\\|\mathbf{m}|=n-1}} \frac{1}{\prod_{i=1}^{k} m_i} + \frac{(n-1)!}{(k-1)!}\sum_{\substack{\mathbf{m}\geq\mathbf{1}_{k-1}\\|\mathbf{m}|=n-1}} \frac{1}{\prod_{i=1}^{k-1} m_i}.$$

Multiplying both sides by $\frac{k!}{(n-1)!}$ we obtain

$$\sum_{\substack{\mathbf{m}\geq\mathbf{1}_k\\|\mathbf{m}|=n}} \frac{n}{\prod_{i=1}^{k} m_i} = \sum_{\substack{\mathbf{m}\geq\mathbf{1}_k\\|\mathbf{m}|=n-1}} \frac{n-1}{\prod_{i=1}^{k} m_i} + \sum_{\substack{\mathbf{m}\geq\mathbf{1}_{k-1}\\|\mathbf{m}|=n-1}} \frac{k}{\prod_{i=1}^{k-1} m_i},$$

which is true from Lemma A.1.1. ■

Theorem A.1.4. *The Stirling numbers of the second kind, $\mathcal{S}_2(n,k)$, with $n,k \geq 0$ are given by*

$$\mathcal{S}_2(n,k) = \frac{n!}{k!} \sum_{\substack{\mathbf{m} \geq \mathbf{1}_k \\ |\mathbf{m}|=n}} \frac{1}{\prod_{i=1}^{k} m_i!}. \tag{A.4}$$

□

Proof. We have to prove that the right hand side of Equation (A.4) has the properties of Definition 5.3.2.

Consider $n,k = 0$, the initial condition is satisfied since the only valid argument of the sum is the 0-tuple. For both $n > 0$, $k = 0$ and $n = 0$, $k > 0$ there are no valid arguments in the sum, which results in zero and those initial conditions are also satisfied.

For the remaining values, $n,k > 0$, we have to show that they follow the recurrence relation, that is,

$$\frac{n!}{k!} \sum_{\substack{\mathbf{m} \geq \mathbf{1}_k \\ |\mathbf{m}|=n}} \frac{1}{\prod_{i=1}^{k} m_i!} = k\frac{(n-1)!}{k!} \sum_{\substack{\mathbf{m} \geq \mathbf{1}_k \\ |\mathbf{m}|=n-1}} \frac{1}{\prod_{i=1}^{k} m_i!} + \frac{(n-1)!}{(k-1)!} \sum_{\substack{\mathbf{m} \geq \mathbf{1}_{k-1} \\ |\mathbf{m}|=n-1}} \frac{1}{\prod_{i=1}^{k-1} m_i!}.$$

Multiplying both sides by $\frac{k!}{(n-1)!}$ we obtain

$$\sum_{\substack{\mathbf{m} \geq \mathbf{1}_k \\ |\mathbf{m}|=n}} \frac{n}{\prod_{i=1}^{k} m_i!} = \sum_{\substack{\mathbf{m} \geq \mathbf{1}_k \\ |\mathbf{m}|=n-1}} \frac{k}{\prod_{i=1}^{k} m_i!} + \sum_{\substack{\mathbf{m} \geq \mathbf{1}_{k-1} \\ |\mathbf{m}|=n-1}} \frac{k}{\prod_{i=1}^{k-1} m_i!},$$

which is true from Lemma A.1.2. ■

Lemma 5.3.6. *For $\mathbf{m}, \mathbf{M} \geq \mathbf{0}_k$, with $k \geq 0$, we have that*

$$\sum_{\substack{\overline{\mathbf{m}} \geq \mathbf{1}_{|\mathbf{m}|} \\ \overline{\mathbf{m}}\mathbf{m}=\mathbf{M}}} \frac{1}{\prod_{i=1}^{|\mathbf{m}|} \overline{m}_i} = \prod_{i=1}^{k} \frac{m_i!}{M_i!} \mathcal{S}_1(M_i, m_i). \tag{5.22}$$

□

Proof. We can break the multi-index $\overline{\mathbf{m}}$ with $|\mathbf{m}|$ elements into the set of multi-indexes $(\overline{\mathbf{m}}^i)$, with $1 \leq i \leq k$, such that each $\overline{\mathbf{m}}^i$ has m_i elements and is associated with M_i. In particular, this means that $\overline{\mathbf{m}}\mathbf{m} = \mathbf{M}$ becomes $|\overline{\mathbf{m}}^i| = M_i$, for every i with $1 \leq i \leq k$. Using this, the left hand side of Equation (5.22) can be written as the product

$$\prod_{i=1}^{k} \left(\sum_{\substack{\overline{\mathbf{m}}^i \geq \mathbf{1}_{m_i} \\ |\overline{\mathbf{m}}^i|=M_i}} \frac{1}{\prod_{j=1}^{m_i} \overline{m}^i_j} \right).$$

The result comes directly from applying Theorem A.1.3. ■

Lemma 5.3.7. *For* $\mathbf{m}, \mathbf{M} \geq \mathbf{0}_k$*, with* $k \geq 0$*, we have that*

$$\sum_{\substack{\overline{\mathbf{m}} \geq \mathbf{1}_{|\mathbf{m}|} \\ \overline{\mathbf{m}}\mathbf{m} = \mathbf{M}}} \frac{1}{\prod_{i=1}^{|\mathbf{m}|} \overline{m}_i!} = \prod_{i=1}^{k} \frac{m_i!}{M_i!} \mathcal{S}_2(M_i, m_i). \tag{5.23}$$

□

Proof. Using the exact same approach as in the proof of Lemma 5.3.6, the left hand side of Equation (5.23) can be written as the product

$$\prod_{i=1}^{k} \left(\sum_{\substack{\overline{\mathbf{m}}^i \geq \mathbf{1}_{m_i} \\ |\overline{\mathbf{m}}^i| = M_i}} \frac{1}{\prod_{j=1}^{m_i} \overline{m}^i_j!} \right).$$

The result comes directly from applying Theorem A.1.4. ■

Lemma A.1.5. *For* $n \geq 0$*, we have that*

$$\sum_{k \geq 1} (-1)^k \mathcal{S}_1(n,k) = \begin{cases} -1 & \text{if } n = 1, \\ 0 & \text{otherwise}. \end{cases} \tag{A.5}$$

□

Proof. The first cases can easily be verified from inspection. Note that the sum is finite since $\mathcal{S}_1(n,k) = 0$ when $k > n$. For $n = 0$ this is a zero sum and for $n = 1$ we have $(-1)\mathcal{S}_1(1,1) = -1$. The remaining terms are proven by induction. Assume the sum to be zero for $n > 1$. We expand $\mathcal{S}_1(n+1,k)$ according to its recurrence relation

$$\begin{aligned} \sum_{k \geq 1} (-1)^k \mathcal{S}_1(n+1,k) &= \sum_{k \geq 1} (-1)^k \left[n\mathcal{S}_1(n,k) + \mathcal{S}_1(n,k-1) \right] \\ &= n \sum_{k \geq 1} (-1)^k \mathcal{S}_1(n,k) + \sum_{k \geq 1} (-1)^k \mathcal{S}_1(n,k-1) \\ &= 0. \end{aligned}$$

The base case of the induction process, $n = 2$, is simply $\mathcal{S}_1(2,2) - \mathcal{S}_1(2,1) = 1 - 1 = 0$. ■

Lemma A.1.6. *For* $n \geq 0$*, we have that*

$$\sum_{k \geq 1} (-1)^k (k-1)! \mathcal{S}_2(n,k) = \begin{cases} -1 & \text{if } n = 1, \\ 0 & \text{otherwise}. \end{cases} \tag{A.6}$$

□

Proof. Firstly, note that for $n = 0$, the sum is trivially 0. Consider now $n > 0$. Then, from Definition 5.3.2

$$\sum_{k\geq 1}(-1)^k(k-1)!\mathcal{S}_2(n,k) = \sum_{k\geq 1}(-1)^k k!\mathcal{S}_2(n-1,k) + \sum_{k\geq 1}(-1)^k(k-1)!\mathcal{S}_2(n-1,k-1).$$

We perform the change of variables $\bar{k} = k-1$ on the second sum of the right hand side, which gives

$$\begin{aligned}\sum_{k\geq 1}(-1)^k(k-1)!\mathcal{S}_2(n,k) &= \sum_{k\geq 1}(-1)^k k!\mathcal{S}_2(n-1,k) + \sum_{\bar{k}\geq 0}(-1)^{\bar{k}+1}\bar{k}!\mathcal{S}_2(n-1,\bar{k}) \\ &= -\mathcal{S}_2(n-1,0),\end{aligned}$$

which is -1 for $n = 1$ and 0 for $n > 1$. ■

Lemma 5.3.8. *For* $\mathbf{M} \geq \mathbf{0}_k$*, with* $k \geq 0$*, we have that*

$$\sum_{\mathbf{m}\geq \mathbf{1}_k}\prod_{i=1}^{k}(-1)^{m_i}\mathcal{S}_1(M_i,m_i) = \begin{cases}(-1)^k & \text{if } \mathbf{M} = \mathbf{1}_k, \\ 0 & \text{otherwise.}\end{cases} \tag{5.24}$$

□

Proof. We can write the left hand side of Equation (5.24) as the product

$$\prod_{i=1}^{k}\sum_{m_i\geq 1}(-1)^{m_i}\mathcal{S}_1(M_i,m_i).$$

Note that although the outer sum of Equation (5.24) looks infinite, it only has a finite number of non-zero elements. Therefore, there are no convergence issues when we do this rearrangement. Consider $k > 0$. If $\mathbf{M} \neq \mathbf{1}_k$, then there will be at least one i with $1 \leq i \leq k$ such that $M_i \neq 1$. From Lemma A.1.5, that term will be zero, which means that the whole product is zero. For $\mathbf{M} = \mathbf{1}_k$ the result is immediate. In the case $k = 0$ we have on the left hand side a sum over one valid index (the 0-tuple) of an empty product, which results in $1 = (-1)^0$. ■

Lemma 5.3.9. *For* $\mathbf{M} \geq \mathbf{0}_k$*, with* $k \geq 0$*, we have that*

$$\sum_{\mathbf{m}\geq \mathbf{1}_k}\prod_{i=1}^{k}(-1)^{m_i}(m_i-1)!\mathcal{S}_2(M_i,m_i) = \begin{cases}(-1)^k & \text{if } \mathbf{M} = \mathbf{1}_k, \\ 0 & \text{otherwise.}\end{cases} \tag{5.25}$$

□

Proof. We can write the left hand side of Equation (5.25) as the product

$$\prod_{i=1}^{k}\sum_{m_i\geq 1}(-1)^{m_i}(m_i-1)!\mathcal{S}_2(M_i,m_i).$$

Using the exact same approach as in the proof of Lemma 5.3.8, the result is straightforward from Lemma A.1.6. ■

Lemma 5.3.10. *Consider a function* ${}^b f_i^{\mathbf{k}}$ *with the properties in Definition 5.3.3, for some* $\mathbf{k} \geq \mathbf{0}_{|T|}$*. For every* $m_{12} \geq 0, \overline{\mathbf{m}} \geq \mathbf{0}_{|\bar{\mathbf{s}}|}$*, such that* $\mathbf{k} = \mathcal{K}(\overline{\mathbf{ms}}) + m_{12}\mathbf{e}_j$*, we have that*

$$ {}^b f_i^{\mathbf{k}}\left(x; \begin{bmatrix} m_{12} \\ \overline{\mathbf{m}} \end{bmatrix}, \begin{bmatrix} w_{j_1} \| w_{j_2} \\ \mathbf{w}_{\bar{\mathbf{s}}} \end{bmatrix}, \begin{bmatrix} x_{j_{12}} \\ \mathbf{x}_{\bar{\mathbf{s}}} \end{bmatrix}\right) = \sum_{\substack{m_1, m_2 \geq 0 \\ m_1 + m_2 = m_{12}}} \binom{m_{12}}{m_1, m_2} {}^b f_i^{\mathbf{k}}\left(x; \begin{bmatrix} m_1 \\ m_2 \\ \overline{\mathbf{m}} \end{bmatrix}, \begin{bmatrix} w_{j_1} \\ w_{j_2} \\ \mathbf{w}_{\bar{\mathbf{s}}} \end{bmatrix}, \begin{bmatrix} x_{j_{12}} \\ x_{j_{12}} \\ \mathbf{x}_{\bar{\mathbf{s}}} \end{bmatrix}\right). \tag{5.26} $$

□

Proof. The proof is by induction. Assume this to be satisfied for $m_{12} = a - 1 \geq 0$. Then, for the case $m_{12} = a$, the left hand side of Equation (5.26) can be written as

$$ {}^b f_i^{\mathbf{k}}\left(x; \begin{bmatrix} a-1 \\ 1 \\ \overline{\mathbf{m}} \end{bmatrix}, \begin{bmatrix} w_{j_1} \| w_{j_2} \\ w_{j_1} \| w_{j_2} \\ \mathbf{w}_{\bar{\mathbf{s}}} \end{bmatrix}, \begin{bmatrix} x_{j_{12}} \\ x_{j_{12}} \\ \mathbf{x}_{\bar{\mathbf{s}}} \end{bmatrix}\right). $$

From assumption, this can be expanded into

$$ \sum_{\substack{m_1, m_2 \geq 0 \\ m_1 + m_2 = a-1}} \binom{a-1}{m_1, m_2} {}^b f_i^{\mathbf{k}}\left(x; \begin{bmatrix} m_1 \\ m_2 \\ 1 \\ \overline{\mathbf{m}} \end{bmatrix}, \begin{bmatrix} w_{j_1} \\ w_{j_2} \\ w_{j_1} \| w_{j_2} \\ \mathbf{w}_{\bar{\mathbf{s}}} \end{bmatrix}, \begin{bmatrix} x_{j_{12}} \\ x_{j_{12}} \\ x_{j_{12}} \\ \mathbf{x}_{\bar{\mathbf{s}}} \end{bmatrix}\right). $$

Using item 2 of Definition 5.3.3 in order to expand over the weight $w_{j_1} \| w_{j_2}$, we get

$$ \sum_{\substack{m_1, m_2 \geq 0 \\ m_1 + m_2 = a-1}} \binom{a-1}{m_1, m_2} \left[{}^b f_i^{\mathbf{k}}\left(x; \begin{bmatrix} m_1 + 1 \\ m_2 \\ \overline{\mathbf{m}} \end{bmatrix}, \begin{bmatrix} w_{j_1} \\ w_{j_2} \\ \mathbf{w}_{\bar{\mathbf{s}}} \end{bmatrix}, \begin{bmatrix} x_{j_{12}} \\ x_{j_{12}} \\ \mathbf{x}_{\bar{\mathbf{s}}} \end{bmatrix}\right) + {}^b f_i^{\mathbf{k}}\left(x; \begin{bmatrix} m_1 \\ m_2 + 1 \\ \overline{\mathbf{m}} \end{bmatrix}, \begin{bmatrix} w_{j_1} \\ w_{j_2} \\ \mathbf{w}_{\bar{\mathbf{s}}} \end{bmatrix}, \begin{bmatrix} x_{j_{12}} \\ x_{j_{12}} \\ \mathbf{x}_{\bar{\mathbf{s}}} \end{bmatrix}\right) \right]. $$

We split this sum into two according to the two terms. Furthermore, we change variables such that $m_1 + 1$ becomes m_1 on the first sum and $m_2 + 1$ becomes m_2 on the second sum. This gives us

$$ \left[\sum_{\substack{m_1 \geq 1 \\ m_2 \geq 0 \\ m_1 + m_2 = a}} \binom{a-1}{m_1 - 1, m_2} + \sum_{\substack{m_1 \geq 0 \\ m_2 \geq 1 \\ m_1 + m_2 = a}} \binom{a-1}{m_1, m_2 - 1} \right] {}^b f_i^{\mathbf{k}}\left(x; \begin{bmatrix} m_1 \\ m_2 \\ \overline{\mathbf{m}} \end{bmatrix}, \begin{bmatrix} w_{j_1} \\ w_{j_2} \\ \mathbf{w}_{\bar{\mathbf{s}}} \end{bmatrix}, \begin{bmatrix} x_{j_{12}} \\ x_{j_{12}} \\ \mathbf{x}_{\bar{\mathbf{s}}} \end{bmatrix}\right). $$

These two sums can be unified into one by appending cases that correspond to zero terms until their index sets match. In particular, in the first sum we can freely append the cases $m_1 = 0$ and in the second sum we can append the cases $m_2 = 0$. Then, we end up with a single sum over the index set $m_1, m_2 \geq 0$, with $m_1 + m_2 = a$.
Then, from the fact that $\binom{n}{m_1,m_2} = \binom{n-1}{m_1-1,m_2} + \binom{n-1}{m_1,m_2-1}$, our expression simplifies into the right hand side of Equation (5.26) and the result applies to $m_{12} = a$.
In base case $m_{12} = 0$, the only valid term in the sum is the one indexed with $m_1, m_2 = 0$. It is clear that for this case the equality is satisfied, which concludes the proof. ■

Lemma 5.3.11. *Consider a family of functions $\{f_i^{\mathbf{k}}\}_{\mathbf{k}\geq \mathbf{0}_{|T|}}$ with the properties in Theorem 5.2.7, for some $\mathbf{k} \geq \mathbf{0}_{|T|}$. For every $m_{12} \geq 0, \overline{\mathbf{m}} \geq \mathbf{0}_{|\overline{\mathbf{s}}|}$, such that $\overline{\mathbf{k}} = \mathcal{K}(\overline{\mathbf{ms}})$, we have that*

$$f_i^{\overline{\mathbf{k}}+m_{12}\mathbf{e}_j}\left(x; \begin{bmatrix} m_{12} \\ \overline{\mathbf{m}} \end{bmatrix}, \begin{bmatrix} w_{j_1} \| w_{j_2} \\ \mathbf{w}_{\overline{\mathbf{s}}} \end{bmatrix}, \begin{bmatrix} x_{j_{12}} \\ \mathbf{x}_{\overline{\mathbf{s}}} \end{bmatrix}\right) \tag{5.27}$$
$$= \sum_{\substack{m_1,m_2\geq 0 \\ m_1,m_2\leq m_{12} \\ m_1+m_2\geq m_{12}}} B(m_1,m_2,m_{12}) f_i^{\overline{\mathbf{k}}+(m_1+m_2)\mathbf{e}_j}\left(x; \begin{bmatrix} m_1 \\ m_2 \\ \overline{\mathbf{m}} \end{bmatrix}, \begin{bmatrix} w_{j_1} \\ w_{j_2} \\ \mathbf{w}_{\overline{\mathbf{s}}} \end{bmatrix}, \begin{bmatrix} x_{j_{12}} \\ x_{j_{12}} \\ \mathbf{x}_{\overline{\mathbf{s}}} \end{bmatrix}\right),$$

where $B(m_1, m_2, m_{12})$ is defined as

$$B(m_1,m_2,m_{12}) := \binom{m_{12}}{m_{12}-m_1, m_{12}-m_2, m_1+m_2-m_{12}}.$$

□

Proof. The proof is by induction. Assume this to be satisfied for $m_{12} = a-1 \geq 0$. Then, for the case $m_{12} = a$, the left hand side of Equation (5.27) can be written as

$$f_i^{\overline{\mathbf{k}}+a\mathbf{e}_j}\left(x; \begin{bmatrix} a-1 \\ 1 \\ \overline{\mathbf{m}} \end{bmatrix}, \begin{bmatrix} w_{j_1} \| w_{j_2} \\ w_{j_1} \| w_{j_2} \\ \mathbf{w}_{\overline{\mathbf{s}}} \end{bmatrix}, \begin{bmatrix} x_{j_{12}} \\ x_{j_{12}} \\ \mathbf{x}_{\overline{\mathbf{s}}} \end{bmatrix}\right).$$

From assumption, this can be expanded into

$$\sum_{\substack{m_1,m_2\geq 0 \\ m_1,m_2\leq a-1 \\ m_1+m_2\geq a-1}} B(m_1,m_2,a-1) f_i^{\overline{\mathbf{k}}+(m_1+m_2+1)\mathbf{e}_j}\left(x; \begin{bmatrix} m_1 \\ m_2 \\ 1 \\ \overline{\mathbf{m}} \end{bmatrix}, \begin{bmatrix} w_{j_1} \\ w_{j_2} \\ w_{j_1} \| w_{j_2} \\ \mathbf{w}_{\overline{\mathbf{s}}} \end{bmatrix}, \begin{bmatrix} x_{j_{12}} \\ x_{j_{12}} \\ x_{j_{12}} \\ \mathbf{x}_{\overline{\mathbf{s}}} \end{bmatrix}\right).$$

Using item 2 of Theorem 5.2.7 in order to expand over the weight $w_{j_1} \| w_{j_2}$, we get

$$\sum_{\substack{m_1,m_2\geq 0\\ m_1,m_2\leq a-1\\ m_1+m_2\geq a-1}} B(m_1,m_2,a-1)\left[f_i^{\bar{\mathbf{k}}+(m_1+m_2+1)\mathbf{e}_j}\left(x;\begin{bmatrix} m_1+1\\ m_2\\ \bar{\mathbf{m}}\end{bmatrix},\begin{bmatrix} w_{j_1}\\ w_{j_2}\\ \mathbf{w}_{\bar{\mathbf{s}}}\end{bmatrix},\begin{bmatrix} x_{j_{12}}\\ x_{j_{12}}\\ \mathbf{x}_{\bar{\mathbf{s}}}\end{bmatrix}\right)\right.$$
$$+ f_i^{\bar{\mathbf{k}}+(m_1+m_2+1)\mathbf{e}_j}\left(x;\begin{bmatrix} m_1\\ m_2+1\\ \bar{\mathbf{m}}\end{bmatrix},\begin{bmatrix} w_{j_1}\\ w_{j_2}\\ \mathbf{w}_{\bar{\mathbf{s}}}\end{bmatrix},\begin{bmatrix} x_{j_{12}}\\ x_{j_{12}}\\ \mathbf{x}_{\bar{\mathbf{s}}}\end{bmatrix}\right)$$
$$\left.+ f_i^{\bar{\mathbf{k}}+(m_1+m_2+2)\mathbf{e}_j}\left(x;\begin{bmatrix} m_1+1\\ m_2+1\\ \bar{\mathbf{m}}\end{bmatrix},\begin{bmatrix} w_{j_1}\\ w_{j_2}\\ \mathbf{w}_{\bar{\mathbf{s}}}\end{bmatrix},\begin{bmatrix} x_{j_{12}}\\ x_{j_{12}}\\ \mathbf{x}_{\bar{\mathbf{s}}}\end{bmatrix}\right)\right].$$

We split this sum into three according to the three terms. Furthermore, we change variables such that m_1+1 becomes m_1 on the first sum, m_2+1 becomes m_2 on the second sum and we apply both changes on the third sum. This gives us

$$\left[\sum_{\substack{m_1\geq 1\\ m_2\geq 0\\ m_1\leq a\\ m_2\leq a-1\\ m_1+m_2\geq a}} B(m_1-1,m_2,a-1)+\sum_{\substack{m_1\geq 0\\ m_2\geq 1\\ m_1\leq a-1\\ m_2\leq a\\ m_1+m_2\geq a}} B(m_1,m_2-1,a-1)\right.$$
$$\left.+\sum_{\substack{m_1,m_2\geq 1\\ m_1,m_2\leq a\\ m_1+m_2\geq a+1}} B(m_1-1,m_2-1,a-1)\right] f_i^{\bar{\mathbf{k}}+(m_1+m_2)\mathbf{e}_j}\left(x;\begin{bmatrix} m_1\\ m_2\\ \bar{\mathbf{m}}\end{bmatrix},\begin{bmatrix} w_{j_1}\\ w_{j_2}\\ \mathbf{w}_{\bar{\mathbf{s}}}\end{bmatrix},\begin{bmatrix} x_{j_{12}}\\ x_{j_{12}}\\ \mathbf{x}_{\bar{\mathbf{s}}}\end{bmatrix}\right).$$

These three sums can be unified into one by appending cases that correspond to zero terms until their index sets match. In particular, in the first sum we can freely append the cases $m_1=0$ and the cases $m_2=a$. Similarly, in the second sum, we can append the cases $m_1=a$ and the cases $m_2=0$. Finally, in the last sum, we can append the cases $m_1=0$, the cases $m_2=0$ and the cases $m_1+m_2=a$. Then, we end up with a single sum over the index set $m_1,m_2\geq 0$, with $m_1,m_2\leq a$, and $m_1+m_2\geq a$. Note that

$$B(m_1-1,m_2,a-1)+B(m_1,m_2-1,a-1)+B(m_1-1,m_2-1,a-1),$$

which is equal to

$$\binom{a-1}{a-m_1,a-m_2-1,m_1+m_2-a}+\binom{a-1}{a-m_1-1,a-m_2,m_1+m_2-a}$$
$$+\binom{a-1}{a-m_1,a-m_2,m_1+m_2-a-1},$$

gives us

$$\binom{a}{a-m_1, a-m_2, m_1+m_2-a} = B(m_1, m_2, a),$$

from the fact that $\binom{n}{m_1,m_2,m_3} = \binom{n-1}{m_1-1,m_2,m_3} + \binom{n-1}{m_1,m_2-1,m_3} + \binom{n-1}{m_1,m_2,m_3-1}$. Therefore, our expression simplifies into the right hand side of Equation (5.27) and the result applies to $m_{12} = a$.
In base case $m_{12} = 0$, the only valid term in the sum is the one indexed with $m_1, m_2 = 0$. It is clear that for this case the equality is satisfied, which concludes the proof. ■

Lemma A.1.7. *For every $m_1, m_2 \in \mathbb{N}$, we have that*

$$\sum_{n=0}^{m_1} (-1)^n \binom{m_1}{n} \binom{n+m_2-1}{m_1-1} = 0. \tag{A.7}$$

□

Proof. The proof is by induction on m_1. Assume Equation (A.7) is valid for a particular $m_1 \geq 1$. Using $\binom{n-1}{k-1} = \binom{n}{k} - \binom{n-1}{k}$, we expand Equation (A.7) into

$$\sum_{n=0}^{m_1} (-1)^n \binom{m_1}{n} \left[\binom{n+m_2}{m_1} - \binom{n+m_2-1}{m_1} \right] = 0$$
$$\sum_{n=0}^{m_1} (-1)^n \binom{m_1}{n} \binom{n+m_2}{m_1} - \sum_{n=0}^{m_1} (-1)^n \binom{m_1}{n} \binom{n+m_2-1}{m_1} = 0.$$

On the first sum we change variables so that $n+1$ becomes n, which gives us

$$\sum_{n=1}^{m_1+1} (-1)^{n-1} \binom{m_1}{n-1} \binom{n+m_2-1}{m_1} - \sum_{n=0}^{m_1} (-1)^n \binom{m_1}{n} \binom{n+m_2-1}{m_1} = 0.$$

We can append the case $n = 0$ on the first sum and the case $n = m_1 + 1$ on the second, which correspond to zero terms. Then, we can merge the two sums back into

$$-\sum_{n=0}^{m_1+1} (-1)^n \left[\binom{m_1}{n-1} + \binom{m_1}{n} \right] \binom{n+m_2-1}{m_1} = 0$$
$$-\sum_{n=0}^{m_1+1} (-1)^n \binom{m_1+1}{n} \binom{n+m_2-1}{m_1} = 0.$$

That is, Equation (A.7) is also valid for $m_1 + 1$. In the base case $m_1 = 1$, the sum gives us $1 - 1 = 0$, which concludes de proof. ■

Lemma 5.3.12. *For every $m_1, m_2 \in \mathbb{N}_0$, we have that*

$$\sum_{\substack{n\geq 1, m_1, m_2\\ n\leq m_1+m_2}} \frac{(-1)^n}{n}\binom{n}{n-m_1, n-m_2, m_1+m_2-n} = \begin{cases} \frac{(-1)^{m_1}}{m_1} & \text{if } m_1 \geq 1, m_2 = 0,\\ \frac{(-1)^{m_2}}{m_2} & \text{if } m_1 = 0, m_2 \geq 1,\\ 0 & \text{otherwise}. \end{cases} \tag{5.28}$$

. □

Proof. For the case $m_1 = 0, m_2 = 0$ the sum is empty, therefore the result is 0. Consider now the case $m_1 \geq 1, m_2 = 0$. Then, the sum consists of only the term indexed with $n = m_1$, which is equal to $\frac{(-1)^{m_1}}{m_1}$. Note that we only need to study the cases with $m_1 \leq m_2$, since the other ones can be trivially deduced thanks to the symmetry of this expression with regard to m_1 and m_2. Therefore, to study the remaining cases $m_1, m_2 \geq 1$, we will now consider the case $1 \leq m_1 \leq m_2$, without loss of generality. We multiply the expression by $\frac{m_1!}{m_1(m_1-1)!}$, where we have that $m_1! = m_1(m_1-1)!$ since we know from assumption that $m_1 > 0$. This gives us

$$\frac{1}{m_1}\sum_{n\geq m_2}^{m_1+m_2}(-1)^n\frac{m_1!}{n(m_1-1)!}\binom{n}{n-m_1, n-m_2, m_1+m_2-n}.$$

We change variables such that n becomes $n+m_2$

$$\frac{1}{m_1}\sum_{n\geq 0}^{m_1}(-1)^{n+m_2}\frac{m_1!}{(n+m_2)(m_1-1)!}\binom{n+m_2}{n+m_2-m_1, n, m_1-n}.$$

This can be further simplified as follows

$$\begin{aligned}
&\frac{(-1)^{m_2}}{m_1}\sum_{n\geq 0}^{m_1}(-1)^n\frac{m_1!}{(n+m_2)(m_1-1)!}\frac{(n+m_2)(n+m_2-1)!}{(n+m_2-m_1)!n!(m_1-n)!}\\
&=\frac{(-1)^{m_2}}{m_1}\sum_{n\geq 0}^{m_1}(-1)^n\frac{m_1!}{n!(m_1-n)!}\frac{(n+m_2-1)!}{(m_1-1)!(n+m_2-m_1)!}\\
&=\frac{(-1)^{m_2}}{m_1}\sum_{n\geq 0}^{m_1}(-1)^n\binom{m_1}{n}\binom{n+m_2-1}{m_1-1},
\end{aligned}$$

which is 0 from Lemma A.1.7. ■

A.2 Intermediate results used in Theorem 5.3.24

Lemma A.2.1 (Theorem 3 of Broder (1984))**.** *The r-Stirling numbers of the first kind are related according to the cross-recurrence formula*

$$\mathcal{S}_1^r(n,k) = r\mathcal{S}_1^{r+1}(n,k+1) + \mathcal{S}_1^{r+1}(n,k), \tag{A.8}$$

for all $n > r \geq 0$ and $k \geq 0$. □

Remark A.2.2. *Note that if one takes the original form of Theorem 3 of Broder (1984) and manipulates it until the present form is reached, the domain obtained would be $n > r > 0$. This can be easily extended for the case $r = 0$ as well, since we have that $\mathcal{S}_1^0(n,k) = \mathcal{S}_1^1(n,k)$ for $n > 0$. This case was missed in the original formula because it corresponded to a singularity.*

Lemma A.2.3. *For $n > r > 0$ and $k > 0$, we have that*

$$\mathcal{S}_1^r(n,k) = (n-r)\mathcal{S}_1^r(n-1,k) + \mathcal{S}_1^{r-1}(n-1,k-1). \tag{A.9}$$

□

Proof. We take the recurrence relation in Definition 5.3.22 and we add and subtract the term $(r-1)\mathcal{S}_1^r(n-1,k)$ to it. That is,

$$\mathcal{S}_1^r(n,k) = (n-1)\mathcal{S}_1^r(n-1,k) - (r-1)\mathcal{S}_1^r(n-1,k) + (r-1)\mathcal{S}_1^r(n-1,k) + \mathcal{S}_1^r(n-1,k-1).$$

The first two terms of the right hand side simplify into $(n-r)\mathcal{S}_1^r(n-1,k)$ while the last two, according to Lemma A.2.1, simplify into $\mathcal{S}_1^{r-1}(n-1,k-1)$ whenever $n > r > 0$ and $k > 0$, which concludes the proof. ■

Lemma A.2.4. *For $r, N, k \geq 0$, we have that,*

$$\sum_{n=0}^{N} \frac{\mathcal{S}_1^r(n+r,k+r)}{n!} = \frac{\mathcal{S}_1^{r+1}(N+r+1,k+r+1)}{N!}. \tag{A.10}$$

□

Proof. The proof is by induction. Assume Equation (A.10) to be satisfied for some $N = a-1 \geq 0$. Then,

$$\begin{aligned}
\sum_{n=0}^{a} \frac{\mathcal{S}_1^r(n+r,k+r)}{n!} &= \sum_{n=0}^{a-1} \frac{\mathcal{S}_1^r(n+r,k+r)}{n!} + \frac{\mathcal{S}_1^r(a+r,k+r)}{a!} \\
&= \frac{\mathcal{S}_1^{r+1}(a+r,k+r+1)}{(a-1)!} + \frac{\mathcal{S}_1^r(a+r,k+r)}{a!} \\
&= \frac{1}{a!}\left[a\mathcal{S}_1^{r+1}(a+r,k+r+1) + \mathcal{S}_1^r(a+r,k+r)\right].
\end{aligned}$$

Consider the change of variables $\bar{n} := a+r+1$, $\bar{r} = r+1$ and $\bar{k} = k+r+1$. Then, this becomes

$$\frac{1}{a!}\left[(\bar{n}-\bar{r})\mathcal{S}_1^{\bar{r}}(\bar{n}-1,\bar{k}) + \mathcal{S}_1^{\bar{r}-1}(\bar{n}-1,\bar{k}-1)\right].$$

Note that the prerequisites for applying Lemma A.2.3 are satisfied. That is, $\bar{n} > \bar{r}$, $\bar{r} > 0$ and $\bar{k} > 0$ correspond to $a+r+1 > r+1$, $r+1 > 0$ and $k+r+1 > 0$ respectively. Therefore, this simplifies into

$$\frac{\mathcal{S}_1^{\bar{r}}(\bar{n},\bar{k})}{a!} = \frac{\mathcal{S}_1^{r+1}(a+r+1,k+r+1)}{a!},$$

which concludes the induction step. For the base case $N = 0$, we have that $\mathcal{S}_1^r(r,k+r) = \mathcal{S}_1^{r+1}(r+1,k+r+1)$, which is always satisfied since $\delta_{r,k+r} = \delta_{r+1,k+r+1}$. ■

Lemma A.2.5. *For $n \geq r \geq 0$ and $k \geq 0$, we have that*

$$\sum_{p=k}^{n-r} \binom{n-p}{r} \frac{\mathcal{S}_1(p,k)}{p!} = \frac{\mathcal{S}_1^{r+1}(n+1,k+r+1)}{(n-r)!}. \tag{A.11}$$

□

Proof. Consider $n = r$. Then, the expression becomes

$$\sum_{p=k}^{0} \binom{n-p}{n} \frac{\mathcal{S}_1(p,k)}{p!} = \frac{\mathcal{S}_1^{n+1}(n+1,k+n+1)}{0!}.$$

If $k = 0$, the left hand side is $\binom{n}{n}\frac{\mathcal{S}_1(0,0)}{0!} = 1$. If $k > 0$, the sum is empty so it is 0. The generalized Stirling number on the right simplifies into $\delta_{n+1,k+n+1}$, which is one if $k = 0$ and zero if $k > 0$. Therefore, equality is achieved for all $k \geq 0$.
Consider now $r = 0$. Then, we have

$$\sum_{p=k}^{n} \binom{n-p}{0} \frac{\mathcal{S}_1(p,k)}{p!} = \frac{\mathcal{S}_1^{1}(n+1,k+1)}{n!}.$$

If $n \geq k$, this reduces to Lemma A.2.4 (note that the missing indexes of the sum correspond to zero terms). If $n < k$ then the left hand side is an empty sum and the Stirling number on the right is zero. The remaining cases that we have to prove are $n > r > 0$, $k \geq 0$, which we prove by induction. Assume Equation (A.11) is satisfied for all (n,r,k) such that $n = a-1$ with $n \geq r \geq 0$ and $k \geq 0$. We now prove that it is satisfied for the cases (a,r,k) with $a > r > 0$ and $k \geq 0$. Note that we have $a - p \geq 1$ for all p in the sum due to the fact that $r > 0$ from assumption. Therefore, we can split Equation (A.11) into

$$\sum_{p=k}^{a-r} \left[\binom{a-p-1}{r-1} + \binom{a-p-1}{r}\right] \frac{\mathcal{S}_1(p,k)}{p!}$$
$$= \sum_{p=k}^{(a-1)-(r-1)} \binom{(a-1)-p}{r-1} \frac{\mathcal{S}_1(p,k)}{p!} + \sum_{p=k}^{(a-1)-r} \binom{(a-1)-p}{r} \frac{\mathcal{S}_1(p,k)}{p!}.$$

Note that the cases $(a-1,r-1,k)$ and $(a-1,r,k)$ satisfy the assumption. That is, $a-1 \geq r-1 \geq 0$

and $a-1 \geq r \geq 0$ are true if $a > r > 0$. We can apply Equation (A.11) to those cases and obtain

$$\frac{1}{(a-r)!}\left[\mathcal{S}_1^r(a,k+r)+(a-r)\mathcal{S}_1^{r+1}(a,k+r+1)\right].$$

From Lemma A.2.3, this simplifies into the right hand side of what we want to prove as long as $a+1 > r+1$, $r+1 > 0$ and $k+r+1 > 0$, which are all satisfied under the current assumptions. The base case $n=0$ has $r=0$, since $n \geq r \geq 0$. This was already covered by the previous cases $n=r$ and $r=0$. ■

Lemma A.2.6. *For $n \geq r \geq 0$ and $k \geq 0$, we have that*

$$\sum_{\substack{\mathbf{m}\geq\mathbf{1}_k\\|\mathbf{m}|\leq n-r}} \frac{1}{\prod_{i=1}^k m_i}\binom{n-|\mathbf{m}|}{r} = \frac{k!}{(n-r)!}\mathcal{S}_1^{r+1}(n+1,k+r+1). \tag{A.12}$$

. □

Proof. Split the sum in the left hand side into the two sums

$$\begin{aligned}\sum_{p=k}^{n-r}\sum_{\substack{\mathbf{m}\geq\mathbf{1}_k\\|\mathbf{m}|=p}} \frac{1}{\prod_{i=1}^k m_i}\binom{n-|\mathbf{m}|}{r} &= \sum_{p=k}^{n-r}\sum_{\substack{\mathbf{m}\geq\mathbf{1}_k\\|\mathbf{m}|=p}} \frac{1}{\prod_{i=1}^k m_i}\binom{n-p}{r}\\ &= \sum_{p=k}^{n-r}\binom{n-p}{r}\sum_{\substack{\mathbf{m}\geq\mathbf{1}_k\\|\mathbf{m}|=p}} \frac{1}{\prod_{i=1}^k m_i}.\end{aligned}$$

From Theorem A.1.3, this is

$$k!\sum_{p=k}^{n-r}\binom{n-p}{r}\frac{\mathcal{S}_1(p,k)}{p!}.$$

the result is now straightforward from Lemma A.2.5. ■

Lemma A.2.7. *For $n \geq k \geq 0$, we have that,*

$$\sum_{\substack{\mathbf{m}\geq\mathbf{1}_k\\|\mathbf{m}|\leq n}} 1 = \binom{n}{k}. \tag{A.13}$$

□

Proof. The proof is by induction. Assume that the result applies for a given $n \geq 0$ and all k such that $0 \leq k \leq n$. We can split the following sum

$$\sum_{\substack{\mathbf{m}\geq\mathbf{1}_k\\|\mathbf{m}|\leq n+1}} 1 = \sum_{\substack{\mathbf{m}\geq\mathbf{1}_k\\|\mathbf{m}|\leq n}} 1 + \sum_{\substack{\mathbf{m}\geq\mathbf{1}_k\\|\mathbf{m}|=n+1}} 1.$$

If $k \geq 1$, we can rearrange the last sum so that we obtain

$$\sum_{\substack{\mathbf{m}\geq\mathbf{1}_k \\ |\mathbf{m}|\leq n+1}} 1 = \sum_{\substack{\mathbf{m}\geq\mathbf{1}_k \\ |\mathbf{m}|\leq n}} 1 + \sum_{\substack{\mathbf{m}\geq\mathbf{1}_{k-1} \\ |\mathbf{m}|\leq n}} 1.$$

From assumption, whenever $k \leq n$, this simplifies into $\binom{n}{k} + \binom{n}{k-1} = \binom{n+1}{k}$. To complete the induction step we now prove the remaining cases $k = 0$ and $k = n+1$. For the first one, the only valid index is the 0-tuple so the sum is always $1 = \binom{n+1}{0}$. For the second one the only valid index is the (n+1)-tuple of all ones so the sum is always $1 = \binom{n+1}{n+1}$. Therefore, the result applies to $n+1$ and all k such that $0 \leq k \leq n+1$.
In the base case $n, k = 0$, the only valid index is again the 0-tuple and the sum gives us $1 = \binom{0}{0}$. ■

Lemma A.2.8. *For $\mathbf{M} \geq \mathbf{1}_k$, with $n \geq |\mathbf{M}|$ and $k \geq 0$, we have that*

$$\sum_{\substack{\mathbf{m}\geq\mathbf{M} \\ |\mathbf{m}|\leq n}} \prod_{i=1}^{k} \binom{m_i - 1}{M_i - 1} = \binom{n}{|\mathbf{M}|}. \tag{A.14}$$

□

Proof. Using Lemma A.2.7, the left hand side can be expanded into

$$\sum_{\substack{\mathbf{m}\geq\mathbf{M} \\ |\mathbf{m}|\leq n}} \prod_{i=1}^{k} \sum_{\substack{\overline{\mathbf{m}}_i\geq\mathbf{1}_{M_i-1} \\ |\overline{\mathbf{m}}_i|\leq m_i-1}} 1.$$

The inner sum can be rearranged such that we get

$$\sum_{\substack{\mathbf{m}\geq\mathbf{M} \\ |\mathbf{m}|\leq n}} \prod_{i=1}^{k} \sum_{\substack{\overline{\mathbf{m}}_i\geq\mathbf{1}_{M_i} \\ |\overline{\mathbf{m}}_i|= m_i}} 1.$$

Distributing the product over the inner sum gives us

$$\sum_{\substack{\mathbf{m}\geq\mathbf{M} \\ |\mathbf{m}|\leq n}} \sum_{\substack{\overline{\mathbf{m}}_1\geq\mathbf{1}_{M_1} \\ \cdots \\ \overline{\mathbf{m}}_k\geq\mathbf{1}_{M_k} \\ |\overline{\mathbf{m}}_1|=m_1 \\ \cdots \\ |\overline{\mathbf{m}}_k|=m_k}} 1.$$

Note that we can completely remove the dependence on $\mathbf{m}$ in this expression. In particular, note that for every i such that $1 \leq i \leq k$, we have that $\overline{\mathbf{m}}_i \geq \mathbf{1}_{M_i}$. Then, $|\overline{\mathbf{m}}_i| \geq M_i$. Since we also have that $|\overline{\mathbf{m}}_i| = m_i$, this implies that $m_i \geq M_i$ for all i. Therefore, the expression $\mathbf{m} \geq \mathbf{M}$ is redundant. Moreover, we can replace $|\mathbf{m}| \leq n$ by $|\overline{\mathbf{m}}_1| + \ldots + |\overline{\mathbf{m}}_k| \leq n$. Defining $\overline{\mathbf{m}}$ as the concatenation of

$\overline{\mathbf{m}}_1, \ldots, \overline{\mathbf{m}}_k$, the expression simplifies into

$$\sum_{\substack{\overline{\mathbf{m}} \geq \mathbf{1}_{|\mathbf{M}|} \\ |\overline{\mathbf{m}}| \leq n}} 1.$$

Since $n \geq |\mathbf{M}|$ from assumption and $|\mathbf{M}| \geq k \geq 0$, the result is now immediate from applying Lemma A.2.7 again. ■

Lemma 5.3.25. *For* $\mathbf{M} \geq \mathbf{1}_k$*, with* $n \geq |\mathbf{M}|$ *and* $k, r \geq 0$*, we have that*

$$\sum_{\substack{\mathbf{m} \geq \mathbf{M} \\ \mathbf{p} \geq \mathbf{1}_r \\ |\mathbf{m}|+|\mathbf{p}| \leq n}} \prod_{i=1}^{k} \binom{m_i - 1}{M_i - 1} \prod_{j=1}^{r} \frac{1}{p_j} = \frac{r!}{(n - |\mathbf{M}|)!} \mathcal{S}_1^{|\mathbf{M}|+1}(n+1, r+|\mathbf{M}|+1). \tag{5.34}$$

□

Proof. We first split the sum of the left hand side and reorganize it as

$$\sum_{\substack{\mathbf{p} \geq \mathbf{1}_r \\ |\mathbf{p}| \leq n - |\mathbf{M}|}} \frac{1}{\prod_{j=1}^{r} p_j} \left[\sum_{\substack{\mathbf{m} \geq \mathbf{M} \\ |\mathbf{m}| \leq n - |\mathbf{p}|}} \prod_{i=1}^{k} \binom{m_i - 1}{M_i - 1} \right].$$

Using Lemma A.2.8, this simplifies into

$$\sum_{\substack{\mathbf{p} \geq \mathbf{1}_r \\ |\mathbf{p}| \leq n - |\mathbf{M}|}} \frac{1}{\prod_{j=1}^{r} p_j} \binom{n - |\mathbf{p}|}{|\mathbf{M}|}.$$

The result now follows from Lemma A.2.6. ■