1

Structural Analysis with Algebra

To find bonds, patterns, and stories that tell us that we are not just product of random.

1.1 Preliminaries

We start with the main rationale behind the structural study of networks and social phenomena, and this initial Chapter provides a basic language and vocabulary required for diverse types of algebraic analyses of social networks. Much of the mathematical formulae are based on classic and standard books about algebraic combinatorics like Kim and Roush (1983), Dunn and Hardegree (2001), or Maddux (2006); within the social networks the literature includes Boyd (1991), Freeman (1992), Pattison (1993), Wasserman and Faust (1994), and Degenne and Forsé (1999).

The arrangement of actors in a social network can be expressed in mathematical terms with the fundamentals from set theory. A *set,* which is the basic structure upon which all other structures are built, is a collection of objects called the *elements* or *members* of the set. Sets can furthermore be described by using what is called the "set build notation." For example, $\{x \mid x > 0 \text{ and } x < 4\}$ reads "the set of all x such that x is greater than zero and less than four." This would be $x = \{1, 2, 3\}$ if x is a member of the set of natural numbers, written as $x \in \mathbb{N}$, and where $\mathbb{N}$ in this case describes completely the set.

In order to avoid the so-called Russell's paradox, which states that a "set of all sets which do not contain themselves does not exist," it is necessary to define the universe of the element in question, and $x \in \mathbb{N}$ does in this example, otherwise if $x \notin \mathbb{N}$ (i.e. in the opposite situation) it should have been added to the expression $x \in X$ for instance where X represents the universe.

Curls and angles are used in the notation for unordered sets and parentheses for the ordered ones. A finite set is called a *list* and we refer to the set as a *family* of objects (rather than merely a collection) when the set sequence is important. Basic operations on sets are the union [resp. intersection], which for sets A and B are denoted by $A \cup B$ [resp. $A \cap B$], and where $A \cup B = \{x \mid x \in A \text{ or } x \in B\}$ [resp. $A \cap B = \{x \mid x \in A \text{ and } x \in B\}$]; and also complement $A^{\complement} = \{x \in X \mid x \notin A\}$.

Algebraic Analysis of Social Networks: Models, Methods and Applications using R,
First Edition. J. A. R. Ostoic. Companion website: www.wiley.com/go/ostoic/algebraicanalysis.

To define a relation in terms of sets, an ordered pair such as (x, y) refers to a directed linkage from an element x to an element y, where $x \in X$ and $y \in Y$. The overall relation set results from the Cartesian product of the sets X and Y, which is the set $X \times Y$ of all ordered pairs (x, y) from it, and this product set is said to be the context of the relation.

A binary relation R between X and Y is then defined as a triple (X, R, Y), where $R \subseteq X \times Y$; and '$\subseteq$' means "is a subset of." Since the relation is defined on an ordered pair, that is a pair of objects with an order associated with them; then actually the relation is a "binary" relation. In the text, however, the term "relation" refers to a relationship with a binary operation, and this to avoid confusion with a special type of tie having just the binary values 0 and 1.

In set theory, X is known as the *domain*, and Y as the *codomain* of a relation. When the domain and the codomain are equal, the ties occur among a single set of entities, which typically are the social actors, and a hence a one-mode (social) network is defined as a domain with a set of relations on such domain. However, sometimes the entire social system needs to be described by more than a single set of entities, and the domain and the codomain result not being equal as with two-mode network data.

After this short preamble, we are able to define in formal terms a social network as a *relational system*:

$$\mathscr{X} = \langle\ \mathbf{X},\ \mathscr{G},\ \mathbf{A}\ \rangle.$$

Each element in this triple stands for a different type of representation of the social network ranging from a concrete account of the system in the first element, to more abstracts versions of the structure in the other two elements.

Loosely speaking, the term "relational system" constitutes the widest notion to refer to diverse representation forms of social networks. We will see later on that this concept applies as well to different types of social networks, and other structures that are derived from these social systems. However, first we take a closer look at each of the above components in $\mathscr{X}$.

In the relational system, $\mathbf{X}$ represents the structure of the simplest social network and comprises a triple of sets:

$$\mathbf{X} = \langle\ \mathscr{N},\ \mathscr{M},\ \mathscr{E}\ \rangle,$$

which is a collection of social actors $\mathscr{N} = \{\ i \mid i \text{ is an actor }\}$, together with a set of ties $\mathscr{E} = \{\ \langle i,j \rangle \mid i \text{ 'has a tie to' } j\ \}$. For one-mode networks $\mathscr{M} = \mathscr{N}$, and for two-mode systems $\mathscr{M} = \{\ i \mid i \text{ is an event }\}$, $\mathscr{M} \neq \mathscr{N}$. This simple structure of the network is called an *algebraic structure* (Freeman, 1992) or simply a *network* (Shier, 1991), and thus $\mathbf{X}$ represents the empirical structure of the social network.

1.2 Graphs

1.2.1 Graphs and Digraphs

The empirical structure can be described in a more abstract way by using the elements of graph theory (Harary, 1994; König, 1936). Hence, a network can similarly be represented by a graph $\mathscr{G}$ where the actors are depicted as points in the graph, and lines connecting the points correspond to the ties between the actors:

$$\mathscr{G} = \langle\ \mathscr{N},\ \mathscr{M},\ \mathscr{E}\ \rangle.$$

In graph theoretic terms, the points are referred to as *vertices* or *nodes*, and the lines as *edges*. When the ties have direction, the graph is directed, $\mathscr{G}^d$, and it is called a *digraph*. $\langle i,j \rangle$ then becomes an ordered pair of nodes, i.e. (i,j), and the directed edges are depicted in the digraph as arcs pointing the direction of the tie. In consequence, a digraph is a labelled graph with a family of relations, as opposed to an unlabeled graph with just a list of relations.

A graph is considered as *complete* when each pair of nodes is joined by an edge; otherwise the graph is not complete. Although for convenience in this case both the sets of actors and ties, and the sets of points and lines are formally represented by $\mathscr{N}$ and $\mathscr{E}$, in principle different characters should be assign the level of abstraction. Nevertheless, for our purpose a social network comprises a domain that is a collection of social entities, $\mathscr{N} = \{ n_1, n_2, \ldots, n_g \}$, $\mathscr{M} = \{ m_1, m_2, \ldots, m_h \}$, and a set of relations, $\mathscr{E} = \{ e_1, e_2, \ldots, e_t \}$, being g as the number of actors or nodes in $\mathscr{N}$, h the number of events in $\mathscr{M}$, and t as the total edges or ties in $\mathscr{E}$ with $t = g + h$.

1.2.2 Multigraphs

Yet until now all these formal representations are for a *simple* social network, and this is because a single type of relation is linking the set of actors in the system. However, social networks are far more complex and typically there is more than one class of relations playing on the same set of actors. To capture the multiplicity of ties, this extra information requires additional notation.

A network having more than one type of relation is called a *multiple* or *multiplex network* and its corresponding graph is a multiple graph or *multigraph*:

$$\mathscr{G}^+ = \langle\, \mathscr{N},\ \mathscr{E},\ \mathscr{R} \,\rangle$$

where $\mathscr{R}$ represents a collection of relations whose elements are $R_1, R_2, \ldots, R_r$, where r is total number of relational types R in the social system. Each relational type then is represented by a distinct set of edges in $\mathscr{E}$.

1.2.3 Signed Graph

Special types of multiplex networks are systems having edges attached with a "sign" or a "valence" to capture the ties with an opposite sense. This type of ties represents affective ties such as "liking" or "disliking," or instrumental relations like "cooperation" and "competition" among economic organizations for instance.

A graph with lines having different valences or signs on the edges constitutes a *signed graph* that is defined as:

$$\mathscr{G}^\sigma = \langle\, \mathscr{N},\ \mathscr{R}, \mathscr{V} \,\rangle.$$

With $\mathscr{G}^\sigma$ there is a set $\mathscr{V} = \{ \upsilon_1, \upsilon_2, \ldots, \upsilon_l \}$ of *valences* attached to $\mathscr{R}$, and where υ_l stands for the total number of valences that a tie or a line can possible bear. For instance, when a relationship is either *positive* or *negative* then l equals 2. However, the meaning that a relation as labeled as "positive" or "negative" does not imply that one relation is "good" and the other is "bad" in a moral sense; it only indicates that there is a change of a *sign* in the valence of the tie.

Moreover, the absence of a tie is also considered as a different valence, and a relationship can be a mixture of different signs as well. In this latter case, a relation is neither positive nor negative, but rather an *ambivalent* type of tie, which means that the value of l is greater than 2. Note that a signed graph contemplates at least two kinds of ties and therefore a signed graph by definition is not a simple network, but it is rather a special case of a multiplex network structure.

1.2.4 Bipartite Graph

A graph $\mathscr{G}$ can be partitioned into s subsets of nodes $\mathscr{N}_1, \mathscr{N}_2, \ldots, \mathscr{N}_\mathsf{s}$ such that each line in $\mathscr{G}$ are between a node in $\mathscr{N}_i$ and a node in $\mathscr{N}_j$, where $i \neq j$. The graph representing a network with this condition, which is termed as s-partite graph (Wasserman and Faust, 1994; Harary, 1994, (1st ed. 1969)), correspond to a one-mode network.

Hence, networks with a single class of actors or nodes have $\mathsf{s} = 1$, and these are partitionable as well. In this case, however, the partition of the system is made according to rules of a chosen type of equivalence among a single class of network members, which are the actors.

When $\mathsf{s} = 2$, then a two-partite graph or the *two-mode* network represents *affiliation networks* in which there is another class of network members that are called "events". Moreover, "multi-mode" networks are then those systems where $\mathsf{s} \geqslant 2$.

The *bipartite graph* $\mathscr{G}^B$ that is formally defined as

$$\mathscr{G}^B = \langle\ \mathscr{N},\ \mathscr{M},\ \mathscr{E}\ \rangle$$

$\mathscr{G}^B$ is then made of 2 sets of entities, $\mathscr{N}_1 = \mathscr{N}$ and $\mathscr{N}_2 = \mathscr{M}$, which typically stand respectively for actors and events, or something else related to them that is not another actor from the actor set.

Bipartite graphs can be multiplex as well when there is a set of relations $\mathscr{R}$ attached to the structure

$$\mathscr{G}^B = \langle\ \mathscr{N},\ \mathscr{M},\ \mathscr{E},\ \mathscr{R}\ \rangle$$

1.2.5 Valued Graph

When the set of ties in a system carries on a *value* or *weight* score that reflects the strength of the relationship between two connected actors, then the type of system corresponds to a *valued network*, also known as *weighted network*. Weights are typically measured in real numbers or $\mathscr{W} \in \mathbb{R}$, and the relational content of the tie corresponds to a qualitative piece of information.

Valued graphs represent a valued network, and a *valued graph* $\mathscr{G}^V$ has a set of values V attached to $\mathscr{G}$. Formally, this kind of structure is defined as

$$\mathscr{G}^V = \langle\ \mathscr{N},\ \mathscr{E},\ \mathscr{W}\ \rangle$$

with $\mathscr{W} = \{v_1, v_2, ..., v_g\}$, and for two-mode networks v_h as well.

As with bipartite graphs, valued graphs that are multiplex also add a set of relations $\mathscr{R}$ to the structure

$$\mathscr{G}^V = \langle\ \mathscr{N},\ \mathscr{E},\ \mathscr{W},\ \mathscr{R}\ \rangle.$$

1.2.6 Multilevel Graph

Structures that are more complex are found in multilevel graphs that are special types of graphs having two or more related sets of interrelated entities. Formally, a *multilevel graph* $\mathscr{G}^M$ with sets $\mathscr{N}$ and $\mathscr{M}$ and a value set $\mathscr{W}$ is defined as

$$\mathscr{G}^M = \langle\ \mathscr{N}\ , \mathscr{M}\ ,\ \mathscr{E}_{\mathscr{N}},\ \mathscr{E}_{\mathscr{M}},\ \mathscr{E}_{\mathscr{N}\ \times\ \mathscr{M}},\ \mathscr{W}\ \rangle.$$

Hence, multilevel graphs are “a kind of” bipartite graphs where the two sets, actors and events, are or can be related within each other. Typically the ties between the sets are dichotomous and the relationships can have different intensities.

In addition, multilevel graphs that are multiplex supplement a set of relations $\mathscr{R}$ to the structure $\mathscr{G}^{M+}$

$$\mathscr{G}^M = \langle \mathscr{N}, \mathscr{M}, \mathscr{E}_{\mathscr{N}}, \mathscr{E}_{\mathscr{M}}, \mathscr{E}_{\mathscr{N}\times\mathscr{M}}, \mathscr{W}, \mathscr{R} \rangle$$

1.3 Matrices

The third element of the relational system $\mathscr{X}$, is another representation of the social network that is in a matrix form. In this case, **A** is a two-dimensional array size $n \times n$ called an *adjacency matrix*, which is associated with a relation R among the actor set in the network.

$$\mathbf{A} = \langle\ a_{ij}\ \rangle, \qquad i \neq j$$

where a_{ij} records the value of a tie from actor i to actor j on the relation. This is actually the same as saying that (i, j) is an element of $\mathscr{E}$. In this case, since there is a direct tie between two actors, they are considered to be *adjacent* to one another and become neighbor nodes.

In the adjacency matrix, the actors in the rows are the *senders*, whereas those actors in the columns are the *receivers* of the ties. When the ties of the network have no direction then $a_{ij} = a_{ji}$, resulting in a symmetric matrix and in a graph with undirected edges. Often a_{ii} is undefined by design for all actors i, which means that an actor cannot relate to himself. In this case, the entries in the *diagonal* of the matrix or $\mathbf{A}^\alpha$ are set to zero and ignored, and the nodes of the graph have no loops.

For a dichotomous relation—that is when a link either exists or not between two parts—the values of a_{ij} are simply represented by 0’s and 1’s, and the entries in the sociomatrix are defined as:

$$a_{ij} = \begin{cases} 1 & \text{if actor } i \text{ has a relation to actor } j, \\ 0 & \text{otherwise.} \end{cases}$$

1.3.1 Affiliation Matrix

For two-mode network data, the adjacency matrix **A** is called *affiliation matrix* with size $n \times m$. The entry the values for a dichotomous relation in this two-dimensional array are:

$$a_{ij} = \begin{cases} 1 & \text{if actor } i \text{ is affiliated to event } j, \\ 0 & \text{otherwise.} \end{cases}$$

1.3.2 Multiple Relations

For multiplex networks, an adjacency matrix $\mathbf{A}(R)$ is defined for each relation-type R:

$$\mathbf{A}(R) = \langle\, a_{ijR} \,\rangle, \qquad i \neq j$$

where a_{ijR} records the value of a tie from actor i to actor j on relation R. The entries for dichotomous relations in the case of a multiplex network are:

$$a_{ijr} = \begin{cases} 1 & \text{if } r \text{ relates actor } i \text{ to actor } j, \\ 0 & \text{otherwise.} \end{cases}$$

As a result of this, there are r sociomatrices with size $n \times n$, one for each relation-type for all the actors in $\boldsymbol{X}$, and the multigraph $\mathscr{G}^{+}$ contains multiple edges sharing the corresponding endpoints. A three-dimensional array $\mathbf{A}(\mathscr{R})$ of size $n \times n \times r$ can serve to represent a network with multiple relations where every "slice" of the vector of matrices corresponds to a relationship class. Such matrix is arranged and labelled identically in order to have a coherent representation of the entire relational system under study.

1.3.3 Incidence Matrix

The connection between nodes and edges corresponds to an *incidence relation*, and this type of association is recorded in a rectangular matrix $n \times e$ of actors by ties that is called an *incidence matrix* **C** defined as

$$a^{\mathbf{C}}_{ik} = \begin{cases} 1 & \text{if node } i \text{ is related to edge } k, \\ 0 & \text{otherwise.} \end{cases}$$

Undirected and directed graphs are represented by "unoriented" and "oriented" incidence matrices respectively.

A relationship between the incident matrix **C** and the adjacent matrix **A** is

$$\mathbf{C} \circ \mathbf{C}^{T} = \mathbf{A}^{\alpha} - \mathbf{A}$$

where the left side of the expression represents the relational composition of the incidence matrix with its transpose (the two operations are defined later on in this chapter), whereas $\mathbf{A}^{\alpha}$ stands for the diagonal of the adjacent matrix.

1.3.4 Valency Matrix

For signed graphs, a special type of adjacency matrix $\mathbf{A}^{\sigma}$ known as the *valency matrix* is defined in the following manner:

$$\mathbf{A}^{\sigma} = \langle\, a^{\sigma}_{ij} \,\rangle, \qquad i \neq j$$

where a^{σ}_{ij} records the valency v of a tie from i to j, and here the entries are:

$$a^{\sigma}_{ij} = \begin{cases} \mathsf{p} & \text{if actor } i \text{ has a positive relation to actor } j, \\ \mathsf{n} & \text{if actor } i \text{ has a negative relation to actor } j, \\ \mathsf{a} & \text{if actor } i \text{ has an ambivalent relation to actor } j, \\ \mathsf{o} & \text{if there is no tie between } i \text{ and } j. \end{cases}$$

Hence, the network can have up to four valences and two actors or nodes in the system can be p-adjacent, n-adjacent, a-adjacent, or they have any tie in common. This last valence is applied as well when there is a relation of indifference between the actors that are aware of each other.

1.3.5 Different Systems

It might be possible that more than one social network is under study, and in this case there is a separate triple in $\mathscr{X}$ representing each relational system, $\mathscr{X}^+ = \mathscr{X}_1, \mathscr{X}_2, \ldots, \mathscr{X}_t$, being t the total number of networks under study. Then a separate matrix $\mathbf{A}(R\mathscr{X}^+)$ is required for each type of relation in every network:

$$\mathbf{A}(R\mathscr{X}) = \langle\, a_{ijR\mathscr{X}} \,\rangle, \qquad i \neq j$$

where $a_{ijR\mathscr{X}^+}$ records the value of a tie from actor i to actor j in relation R, and in the relational system $\mathscr{X}^+$.

Once more, the entries in $\mathbf{A}(R\mathscr{X})$ for a dichotomous relation can be:

$$a_{ijR\mathscr{X}^+} = \begin{cases} 1 & \text{if } R \text{ relates actor } i \text{ to actor } j \text{ on network } \mathscr{X}^+ \text{ ,} \\ 0 & \text{otherwise.} \end{cases}$$

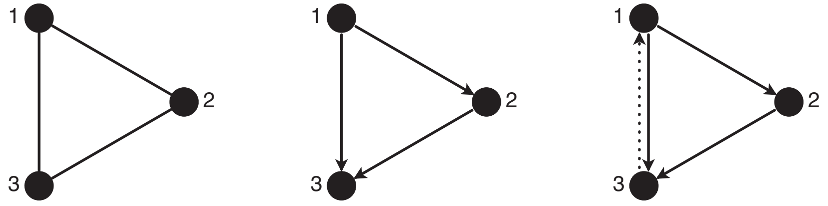
1.3.6 Graph and Matrix Representations

$\mathscr{G}$ and $\mathbf{A}$ usually stand for the social networks and these representation forms are referred as *graph theoretic* and *sociometric* notations by Wasserman and Faust (1994). There are advantages and disadvantages of each kind of representation but it is worthy to mention that sociometric notation permits to inclusion of measurements on actors in the analysis together with different social ties. However, the multiplicity of ties and the combinations of indirect relations can result quickly cumbersome with both the graph theoretic and sociometric notations. For the analysis of different kinds of relations and combinations of these, the algebraic notation results a better alternative and therefore the rest of the chapter has more on algebraic notation.

For instance, Figure 1.1 gives both graph and matrix representations for a configuration made of three connected nodes or a triad. The triad is then given as a simple graph and a symmetric matrix in Figure 1.1a, whereas Figure 1.1b shows a digraph where arcs pointing the direction of the ties from one node to another represent relations and the matrix is in this case asymmetric.

At this point, it is possible to distinguish some network effects like mutuality in dyads, i.e. configurations made of two elements, or in higher level structures a phenomenon like transitivity, which is the tendency that relates two actors that have a common neighbor. Later on, we will see how these particular properties of the configurations and others serve to describe the group structures under study.

The representation in Figure 1.1c characterizes a signed graph where there are two levels in the relations that are represented by directed edges with different shapes. In this case, solid arcs stand for positive relations and a dashed arc points a negative tie, whereas the absence of a tie is represented by o in the signed matrix. Note that if the graph of Figure 1.1c stood for a general directed multigraph, then both the positive and negative relations would have needed a separate adjacency matrix for a complete representation of the system.



	1	2	3
1	0	1	1
2	1	0	1
3	1	1	0

(a) Undirected

	1	2	3
1	0	1	1
2	0	0	1
3	0	0	0

(b) Directed

	1	2	3
1	o	p	p
2	o	o	p
3	n	o	o

(c) Signed

Figure 1.1 Three types of networks with graph and matrix representations. *A shape in $\mathscr{G}$, $\mathscr{G}^d$, and $\mathscr{G}^\sigma$ corresponds to the content in their respective matrices.*

1.4 Chains, Paths, and Other Graph Properties

There are several concepts and properties of graphs and digraphs that are useful for the analysis and characterization of social networks, and the involve systems with either a single type or different types of relations.

We start by defining an alternating sequence of nodes and edges in a graph $\mathscr{G}$:

$$e_i = n_0, e_0, n_1, \dots, e_k, n_k$$

that is called a *chain* when the sequence in $\mathscr{G}$ begins in a node n_0 and ends in node n_k. These are known as the initial and terminal nodes respectively of the sequence; that is $e_i = \langle n_0, n_k \rangle$. On the other hand, if the sequence of nodes and edges is placed in a directed graph $\mathscr{G}^d$, then the arrangement is called a *path* $e_i = (n_0, n_k)$ and the order of the elements is important. A *semipath* in the graph disregards directionality, which implies that $e_i = (n_0, n_k)$ or (n_k, n_0). Not all chains are paths but the converse is always true.

A chain, path, or semipath is called *open* or *simple* if and only if $n_0 \neq n_k$, whereas the sequence is called *closed* when $n_0 = n_k$. A simple closed chain, path, and semipath are called a *circuit*, a *cycle*, and a *semicycle* respectively. Now, k represents the *length* of the sequence, and it is the total number of edge occurrences in the chain, path, or semipath that can be either open or closed.

A graph or a digraph is said to be *connected* if and only if there is a chain, semipath, or path of relations between each pair of nodes in the configuration, which implies that the graph is *unilaterally*, *weakly*, or *strongly* connected respectively. In any of these cases, each node is *reachable* from the other nodes in the graph. As a result, a connected graph with either of the kinds of connectedness just mentioned has a single *component* and no *isolated* nodes, which are defined as nodes having no edges attached in the network. On the other hand, a graph is *disconnected* when it has either more than one component of connected nodes or isolated nodes, or both of them. A trivial graph containing just one node and edge is regarded as a strongly connected graph and not an isolated node.

1.5 Algebra of Relations

Several concepts and properties of graphs and digraphs are useful for the analysis and the characterization of social networks either with single or different types of relations. However, in the case of systems made of multiple relations, a fundamental matter of study is the *algebra* of network relations.

Algebra is a mathematical branch that deals with symbolic relations, and it can be effectively applied in the analysis of systems made of multiple relations. In algebraic notation the relations between actors are written in capital letters instead for subscripts. For instance, if $a_{ijR} = 1$ such that i and j are in $\mathbf{A}$ and $R \subseteq \mathbf{A} \times \mathbf{A}$, means that there exists a relation R between actor i and actor j. If $i \neq j$ in $\mathscr{G}_d$ then actor i "chooses" actor j in a relation, and this implies that $(i,j) \in R$. Such expression can be written just as *iRj* using the infix notation, which is sometimes used in arithmetic where the operator relation is written between the parts or operands on which they act.

1.5.1 Generators and Compounds

When $r > 1$; that is if the network has different kinds of relationships, each relation-type in $\mathscr{G}^+$ constitute a *primitive* or *generator* relation where different operations can be applied to it. This means that in a multiple network is possible to characterize the interaction among relations whenever the appropriated operation is used. As a way to elaborate this idea, we consider each relation type as a *letter* on an *alphabet* $\mathbf{\Sigma}$, which is a finite set of r ties; that is

$$\mathbf{\Sigma} = \{\ R_1, R_2,\ \ldots\ , R_r\ \}.$$

Then one can apply operations on a given set with some defined rules and in this way create a system in the algebraic sense. Hence, an algebraic system or "algebra" is defined as a set of *objects*, together with a set of *operations* on it. For instance, the alphabet and all products over $\mathbf{\Sigma}$ produce an algebraic system known as "free word algebra," which is an unconstrained class of algebra that permits the construction of representations of multiplex social networks in algebraic form.

The *free word algebra* is represented as:

$$\mathbf{W} = \langle\ \mathbf{\Sigma}^*\ ,\ \boldsymbol{F}\ \rangle,$$

where $\mathbf{W}$ stands for the algebraic object, $\mathbf{\Sigma}^*$ for the unconstrained object set, and $\boldsymbol{F}$ for the operations set.

Hence, since in the algebra $\mathbf{W}$ each single relationship is a generator that constitutes a letter of the alphabet of multiple relations, this suggests that the combination of letters will make a "word" or "string" that is a product of the *composition* of relations. Either words can be made from different types of letters or by reusing the same letters where the number of letters determinates the *length* of the word, which in principle can be infinite even just with one generator. However, in a limited population some of these relations will connect precisely the same pair of individual actors, which means that the number of unique compound relations will also be finite (Lorrain and White, 1971; Wasserman and Faust, 1994, p. 495).

It is also possible to "constrain" the free word algebra and limit this object to words until certain length k. This means that we obtain the algebra $\mathbf{W}^k$, which is a truncated version of **W**, and we use for its analysis a *partial algebra* rather than a full algebra. The difference now is that in $\mathbf{W}^k$ the object set in the system is made by the single relationships and by a collection of compound relations up until length k. In either of the two cases, we can define algebraic structures useful for the analysis of multiplex networks depending on the imposition of certain conditions on the strings.

1.6 Operations on Social Networks

Operations defined on pairs of elements from social networks are fundamental in making up the algebraic structures that describe collections of ties in the social systems, and the structure made by these relations can have either a single type of tie or multiple kinds of relationships. Different types of relationships can be differentiated by looking at their patterns of tie overlapping and also through the categorization of indirect relations (White, 1992, p. 95; also Pattison 1993), and where tie overlapping is described by the *intersection* of elements, i.e. $R_1 \cap R_2$, and the categorization of indirect relations by applying the tie composition, respectively.

1.6.1 Binary Operation on Relations

There is natural association between operations and relations where operations are regarded as special sort of relations. Operations are functions that assign elements from a domain X to a codomain Y, and for instance a function f on X and Y will be written as $f : X \rightarrow Y$ using the arrow notation. If we consider the domain of social relations X, the function will have a single input, namely the relationship itself, which is in this case a unary operation written as $f : X \rightarrow X$.

A *binary operation*, on the other hand, is a function that assigns the elements of the Cartesian product of the relations to a third set, that is $f : X \times X \rightarrow X$. This means that while the unary operation has one operand or argument, the binary operation takes two arguments. However, an operation can also have no arguments and it can be a "nullary" operation that produces a constant like the empty set of relations. Thus the set of operations $\boldsymbol{F}$ in **W** has a number of operands that determines its "arity," and where a n-ary operation is a function that assigns an element of a set to each n-tuple of elements of the set. We state this formally by defining an n-ary product R^n on a set $X \mid R^n \subseteq X \times \ldots \times X$ (n times) as the generalization of the direct (aka "Cartesian") product of sets.

Some examples of binary operations are the union and intersection of sets, and there are other binary operations crucial for the analysis of multirelational systems. Before looking at these, we introduce first two important unary operations for the analysis of social networks, which are the complement and the converse of a relation.

In formal terms, the (Boolean) *complement* of a tie R in a system $\boldsymbol{X}$ is defined as:

$$R^{\mathsf{C}} = \{ \langle i,j \rangle \mid j,i \in \boldsymbol{X} \text{ and } \langle j,i \rangle \notin R \}.$$

Hence, the complement of R consists of all ordered pairs of elements of the universe of discourse that do not belong to R (Maddux, 2006, p. 5). For example, if iRj means that "actor j is R related to actor i" then $iR^{\complement}j$ implies that "actors j is *not* R related to actor i."

For directed networks, R^{-1} represents the *converse* of a relation R that is produced by reversing the order of the members of a pair, and this is formally defined as:

$$R^{-1} = \{ \ (i,j) \mid (j,i) \in R \ \}.$$

The converse relation serves to define the asymmetry of ties in $\boldsymbol{X}$, and it is useful to identify passive subjects of relational actions. For instance, if R means "supervising" then R^{-1} will mean "being supervised." Arabie et al. (1978) point out that the converse operation in social relations serves to generate *relational contrast* in the social system, and it is often very useful in the modeling process of its "relational structure", which is the configuration where relations are tied.

Furthermore, the converse relation of an entire network is represented by the *transpose* of the adjacency matrix, $\mathbf{A}^T$, which is obtained by exchanging the matrix rows for its columns. That is, for each a_{ij} in $\mathbf{A}$ we get a_{ji}.

The categorization of indirect relations plays an essential part in the analysis of networks of multiple relations, and this is because it combines relationships that can be primitives or not into indirect links are occurring in the system. For this reason, we are going to take next the composition of ties in a more detailed manner.

1.6.2 Relational Composition

The *relational composition* is a binary operation that is useful in the interpretation of the organizing principles of multiplex networks. The importance of this operation lays in the fact that by combining the different types of a tie occurring in the network, we create an abstract structure that represents the entire relational structure of the system.

A combination of ties then creates an indirect link called a *compound* relation between two nodes in the system through *right multiplication*. The right multiplication procedure adjoins a generator strings to the right of the path label where both equations and orderings—defined later on in this chapter—are preserved. This kind of indirect tie can result either in a relational path or in a chain of relations depending on whether the network is directed or undirected, and it can combine relations at different levels in the system.

The operation of relational composition is denoted by the symbol '$\circ$', and in order to illustrate this operation consider, for example, the relations $R_1 = \mathsf{F}$ and $R_2 = \mathsf{E}$, representing e.g. friendship and enmity. A combination of these link-types makes a new compound relation where a tie $i(\mathsf{F} \circ \mathsf{E})j$ is present if an actor k exists such that $i\mathsf{F}k$ and $k\mathsf{E}j$. This implication can be written by dropping the symbol for the operation of composition as $i(\mathsf{FE})j$ or simply by $i\mathsf{FE}j$, which means that there is a compound relationship made of F and E among actors i and j, and it is read as "actor j is 'the enemy of a friend' of actor i." In this sense, the relational composition involves at least three subjects in the relational path, namely actors i, j, and k in the example, and by convention it reads from right to left.

To have a more formal characterization of the composition operation, let $R_1 \subset X \times Z$ and $R_2 \subset Z \times Y$ (where $\subset$ means "is a proper subset of"). The composition of R_1 and R_2 is a binary relation between X and Y that for all $x \in X$, $y \in Y$ is given by:

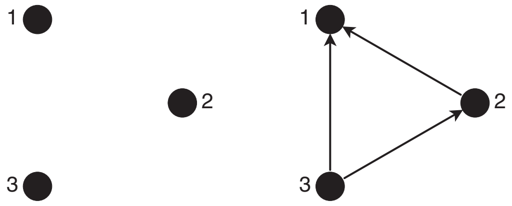
$$R_1 \circ R_2 = \{ \ (x, y) \mid \text{exists } z \in Z \text{ such that } (x, z) \in R_1 \text{ and } (z, y) \in R_2 \ \}.$$

As a result, $R_1 \circ R_2$ denotes the product of R_1 and R_2 or the compound relation between x and y that in this case corresponds to the *matrix product* of the arrays representing R_1 and R_2.

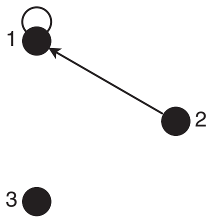
It may be possible for the relational composition that no ordered pairs of actors exist on a particular compound relation. This happens when there are no elements x, y, and z such as $(x, z) \in R_1$ and $(z, y) \in R_2$; thus the relation $R_1 \circ R_2 = \emptyset$, meaning that such compound relation is empty or undefined for x and y. Since we represented collection of relation types in a multiplex network by $\mathscr{R}$, the set of primitive and compound relations are going to be represented by $\mathscr{R}^+$, whereas r^+ will denote the number of elements in $\mathscr{R}^+$.

However, the composition of two operands such as different kinds of relations can be based in other types of operations than the matrix product. For instance, the max − min product is used to create compound relations of weighted or valued ties, but we are going to take a look at these operations when we introduce these types of networks later on.

As illustration, Figure 1.2 presents some operations performed on the graphs depicted in Figure 1.1, and Figure 1.2a, for example, represents the complement of the undirected and loopless graph in Figure 1.1a. The converse of ties in the digraph in Figure 1.1b is shown in Figure 1.2b, and finally Figure 1.2c represents the relational composition of the positive ties and the negative tie in Figure 1.1c where actor 1 ends up having a pair of 2-step paths that combine the two types of valences.



(a) Complement (b) Converse



(c) Composition p ∘ n

Figure 1.2 Different operations on $\mathscr{G}$, $\mathscr{G}^d$, and $\mathscr{G}^\sigma$ from Figure 1.1

1.7 Types and Properties of Relations

We characterized an algebraic structure as a relational system with a family of operations defined on a set where operations are sorts of relations with their own properties and rules that must conform in order to describe properly the structures they made. In this section, we are going to differentiate types of relations in general and their properties for two basic reasons. First properties on relations are fundamental in the characterization of the different types of functions operating in such structures, and second because some specific types of relations with defined properties are building blocks in the establishment of the algebraic structures that represent multiplex networks.

In this sense, we start with three standard relations that have been already introduced either directly or indirectly:

1. the *empty* relation $\varnothing$
2. the *universal* relation U
3. the *identity* relation I

The empty relation is the smallest relation and it is simply the empty set. Since in this case relations are sets of pairs, the empty relation never holds for all the pairs in the set, i.e. $R = \varnothing$. The universal or full relation U is the widest possible relation and it is the Cartesian product of the set, $R = X \times X$, which always holds for all the pairs in set X. Lastly the identity relation I holds for pairs whose first and second elements are identical, that is $R = \{\ (x_i, x_i) \mid \text{for all } x \in X\ \}$.

In a graph format, the identity relation is represented by loops in the nodes, whereas in a matrix form, the empty and the identity relations are the zero-matrix and the identity-matrix respectively; the former with all its entries being zero, and the latter has ones in its main diagonal and zeros elsewhere. The universal relation on the other hand has 1s in all the entries of the matrix including in the diagonal. However, these particular structures assume matrix multiplication in the relational composition, and this is because for matrix addition operation, for example, it is the zero-matrix that represents the identity element.

The three above mentioned standard relations can in turn be characterized in terms of relational composition, and when they are concatenated for instance with a "concrete" relation R, it results in the following identities:

$$\begin{aligned} \varnothing \circ \mathsf{R} &= \varnothing \\ \mathsf{I} \circ \mathsf{R} &= \mathsf{R} \\ \mathsf{U} \circ \mathsf{R} \circ \mathsf{U} &= \mathsf{U}, \qquad \text{iff} \quad \mathsf{R} \neq \varnothing \end{aligned}$$

where "iff" is a shortening of "if and only if," a logical operator that is also represented by the symbol '$\Leftrightarrow$'.

As a result, the product of a relational composition with an empty relation equals the empty set; the identity relation acts as a neutral element in a composition, and finally the universal relation includes all other relations except for the empty relation. Sometimes each one of these relations have the universe attached as $\varnothing_X$, I_X, U_X, R_X, but for simplicity we are going to avoid this when the statements are clear by the context.

It is time now to see some important properties for the existing relation R that are key conditions for making up some algebraic structures useful for the analysis of multiplex networks:

$$\begin{aligned}
\textit{reflexivity} &\Leftrightarrow \mathsf{R} \cup \mathsf{I} \subseteq \mathsf{R} \\
\textit{symmetry} &\Leftrightarrow \mathsf{R}^{-1} = \mathsf{R} \\
\textit{antisymmetry} &\Leftrightarrow \mathsf{R} \cap \mathsf{R}^{-1} = \mathsf{I} \\
\textit{transitivity} &\Leftrightarrow \mathsf{R} \circ \mathsf{R} \subseteq \mathsf{R} \\
\textit{invertibility} &\Leftrightarrow \mathsf{R} \circ \mathsf{R}^{-1} = \mathsf{I}
\end{aligned}$$

Since the identity relation I is included in the reflexive property it means that the element is related to itself. Although the reflexive relation seems a trivial one, it is in fact a consequence of transitivity and reciprocity (Lorrain, 1975, p. 20). Reciprocity is reflected by the symmetric property where a relation equals its inverse, whereas the transitive property implies that the extension of the relation exactly corresponds to the relation itself, and in this sense the notion of transitivity implies that actions of the elements in a system can have “indirect” effects on one another (Berkowitz, 1982).

In the antisymmetric property there is no reciprocity but self-relations can occur. If loops were not allowed by definition, then the relation R is meant to be called *asymmetric* rather than antisymmetric whenever $\mathsf{R} \cap \mathsf{R}^{-1} = \varnothing$. The distinction between antisymmetry and asymmetry in the properties of binary relations serves for the definition of particular types of relations used in the algebraic analysis of relational systems. On the other hand, the asymmetry of social ties indicates that the relationship between a pair of actors has a single direction. Finally, an identity relation or a loop in the graph is produced by the composition of a relation with its converse R^{-1}, which is also called the *inverse* relation of R.

Relations can have one or more of such properties or none of these, and the combination of the properties permit us to identify special types of binary relations relevant for the analysis of different systems. For instance the statement “to be similar to” on a set of people is a sort of relation that is both symmetric and reflexive, and it is known as a *tolerance*. Such kind of relation is symmetric because the relation can go in both directions and it is reflexive because one is similar to him/her/itself.

1.8 Equivalence and Ordering

Two particularly important kinds of relations for the study of social systems are *equivalences* and *ordering* relations, and this is because they serve to classify the actors and relations of the network in different ways. In an equivalence relation two elements are the same in some respect such as “being alike” or “being in the same team” on a set of people.

Ordering relations, on the other hand, make a comparison of lesser to greater elements, i.e. “greater than” or “less than,” which are represented (as seen before) by $>$ or $<$, respectively. Orderings are important to establish hierarchies in the system and also in performing asymmetric clustering (Boyd, 1980), which is a procedure that transforms network data.

1.8.1 Equivalence

The specific properties of the equivalence relation and the related concept of partition of the system are now expressed in mathematical terms. Both concepts are central in the modeling of social networks, since they are used to reduce the complexity of the systems.

Formally, a binary relation R defined on set X is an *equivalence relation* $\equiv$ whenever is:

$$\begin{array}{llll} x \equiv x, & \text{for all} \quad x \in X & & \text{(reflexive)} \\ x \equiv y & \text{implies} \quad y \equiv x & & \text{(symmetric)} \\ x \equiv y & \text{and} \quad y \equiv z \quad \text{imply} & x \equiv z & \text{(transitive)} \end{array}$$

where the expression $x \equiv y \bmod R$ indicates that there is an equivalence relation between x and y under R, and R is a subset of $X \times X$; that is $(x, y) \in R$.

To define the classes of equivalent elements, let $[x]$ denote a subset of X of elements y such that $x \equiv y$. By the reflexivity property we have that $x \in [x]$, and by the symmetry and transitivity properties we have that if $y, z \in [x]$ then $y \equiv z$. This means that $[x]$ is a collection of equivalent elements, which makes it possible to produce a *partition* or a disjoint sets of equal elements in the network that characterizes the "relational structure" in a simplified form. Hence, each equivalence relation establishes an associated partition of the involved elements of the set, and every partition corresponds to a determinate type of equivalent relation.

Formally, a partition π of a set X is any collection $\{ X_i \mid i \in P \}$ of nonempty subsets of X, where π is an indexed set, and which satisfies the following two conditions:

$$\begin{array}{lll} X_i \cap X_j = \varnothing, & \text{for } i,j \in \pi \text{ with } i \neq j & (\textit{mutually exclusive}) \\ X = \bigcup \{ X_i \mid i \in \pi \} & & (\textit{collectively exhaustive}). \end{array}$$

The mutually exclusive condition means that nothing can belong simultaneously to both subsets, and the collectively exhaustive condition means that everything must belong to one subset or the other subset. In other words, a partition of a set X is a decomposition of X into non-overlapping and non-empty subsets or "parts" whose union is all of X and whose intersection is empty.

1.8.2 Partial Order

In the case of ordering relations, special attention is given to the partial ordering since it helps to define the set of distinct relations in a multiplex network represented by an algebraic structure.

The conditions for a *partial order relation* $\leq$ imply that the relation is:

$$\begin{array}{llll} x \leq x, & \text{for all} \quad x \in X & & \text{(reflexive)} \\ x \leq y & \text{and} \quad y \leq x & \text{imply} \quad x = y & \text{(antisymmetric)} \\ x \leq y & \text{and} \quad y \leq z & \text{imply} \quad x \leq z & \text{(transitive)} \end{array}$$

We note that the partial order, as any type of ordering, lacks symmetry, and that is the main difference from the equivalence relation, which is also reflexive and transitive.

A partial ordering on a set determines a *partially ordered set* or *poset* for short, and for a given collection of elements, it is possible to obtain a partial ordered structure, which can be represented by a diagram depicting ordering relation between certain pairs of elements of the set. Often the collection of poset elements makes certain types of algebraic structures with specific characteristics, which allow us to get an insight into the relational system involved. Algebraic structures are covered later on in this chapter and in more depth in the following chapters.

As seen, both the equivalence relations and the orderings have the transitive property, which for equivalence means that if two elements are equivalent to a third one this implies that they are also equivalent between themselves. For an ordering relation the order-preserving condition holds; that is if, for example, x is less than y and y is less than z, then x is less than z; or more formally $x < y$ and $y < z$ imply $x < z$.

Besides transitivity, equivalence relations are also reflexive and symmetric, whereas orderings relations are on the contrary in general irreflexive and asymmetric. There are, however, special types of ordering relations, and for instance a *quasi-order* is a transitive relation that is also reflexive, whereas a quasi-order that is antisymmetric is termed a *partial order*.

Strictly speaking a partial order relation that is reflexive is a “weak” partial order, and with the irreflexive property becomes a “strong” partial order. Partial order differs from a *total* or *lineal* order since a total order relation is said to be connected, complete, and satisfies the “law of trichotomy,” which means that for all $x, y \in X$; either $x > y$, or $y < x$, or $y = x$ holds.

1.8.3 Hierarchy

With the definition of a poset and the equivalence it is possible to define the notion of hierarchy in the network, which is a form for gradation where superordinate and subordinate elements occupy different levels in the systems.

Thus, for a set of relations R, two subsets $X, Y \in R$ form a *hierarchy* whenever one of the following conditions hold:

1. $X \cap Y = \emptyset$
2. $X \cap Y = X \quad \Leftrightarrow \quad Y \leq X$
3. $X \cap Y = Y \quad \Leftrightarrow \quad X \leq Y$

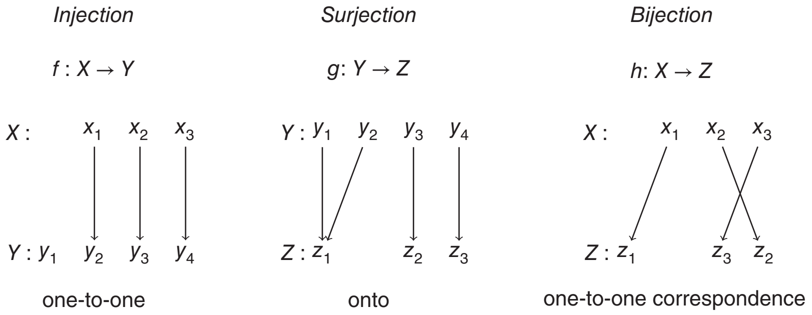
That is, in case one of the subsets contains the other subset, or when these subsets form a partition of the system, which means that they can be pairwise disjoint as well.

1.9 Functions

A function is a particular type of binary relation used in set theory to compare structures in which a quantity in the domain uniquely determines a second quantity in the codomain. The domain is what can go into the function, the codomain is what may possibly come out of it, and what actually comes out of a function is the *range*. Thus each element of the domain is related to a point in the range, like a cartographical map corresponds to a geographical region, and that is why functions are also called “maps” or “mappings”.

For instance, $f : X \to Y$, reads the function f with X as its domain and Y as its range. If for $x \in X, y \in Y, (x, y) \in f$, then $f(x) = y$, where y is called the *image* of x under f and x is the argument or *preimage* of y.

Important classes of functions are:



Thus for each element of the range there is "at most" one pair (x, y) in f, "at least" one pair (y, z) in g, and "exactly" one pair (x, z) in h. More precisely, while in the injective function f elements in the range Y are mapped into by a unique element of the domain X, with the surjective function g there are some elements in the domain Y matching the elements in the range Z, meaning that the image equals its codomain. On the other hand, the bijection is both one-to-one and onto, and in this function there is exactly one element in the domain for every element in the range, which means that the bijective function can be inverted; that is $h^{-1} = Z \rightarrow X$.

A bijection on a finite set as in h is called a *permutation*, and this function is actually the rearrangement of the elements in the set. Permutations are important in the study of cohesive subgroups in social systems and for the reduction of the structure into sets of equivalent actors.

Since functions are special types of binary relations, they can also be "concatenated." In this sense, if $f : X \rightarrow Y$ and $g : f(x) \rightarrow Z$ then $g \circ f : X \rightarrow Z$ by setting $(g \circ f)(x) = g(f(x)), x \in X$. The sequence of f and g in the expression $g \circ f$ is important, because the composition $f \circ g$ might not been even defined. If so, it is usually that $f \circ g \neq g \circ f$, which means that these relations are not *commutative*. However, the composition of functions can be *associative*, and this means that $h \circ (g \circ f) = (h \circ g) \circ f$. In such case the brackets can be omitted and the expression is written as $h \circ g \circ f$, or simply by juxtaposition as hgf.

Commutativity and associativity are axioms or laws that operations and relations conform, and in order to put in formal terms these and other important axioms, we define for all f, g, h of $\boldsymbol{F}$ the following equalities:

Commutativity: $fg = gf$
Associativity: $f(gh) = (fg)h$
Idempotence: $ff = f$
Absorption: $fg = f$ or $fg = g$

Besides commutativity and associativity, the *absorption* law is an identity between two functions or operations whose product equals only one of them, and it is usually present in algebraic structures having an ordering among its elements. In the case of *idempotence* we should note that this axiom is actually a transitive relation when the equality holds, and this means that the operation f can be applied several times to the relation without changing the result.

When idempotence is present in relational composition of a social network, it represents stability or invariance in a system (Boyd, 1991, p. 29). Idempotence is normally applied in relational systems to perform the transitive closure operation, which is a special kind of a transitive relation that is important to find out the connectivity or reachability in the network. Different types of closures are defined in Chapter 2 Algebraic Structures.

On the other hand, commutativity and associativity are properties that are useful in building scalable systems. Associativity enables for instance parallelism in distributed computing, while commutativity allows us to ignore order in the involved data. In the same way, we can apply the principles of commutativity and associativity in the analysis of social networks, where we employ as well as some other special types of functions.

Before introducing other special types of functions useful for the analysis of networks, it is worth mentioning that a commutative diagram can represent compound functions. Commutative diagrams map the compositions of functions and the objects involved in these, and below is the commutative diagram representing relations between the three classes of functions injection, surjection, and bijection introduced before:

$$\begin{tikzcd}
X \arrow[r, "f"] \arrow[dr, "h = f \circ g"'] & Y \arrow[d, "g"] \\
 & Z
\end{tikzcd}$$

1.9.1 Identity and Empty Functions

As in the case of the standard relations, there are also special types of functions. For instance the *identity function* maps each element of a given set to itself, i.e. $f(x) = x, x \in X$; whereas the *empty function* is a unique function from the empty set to a give other set. In this sense, the empty function is a constant that satisfies the definition of "vacously" since the expression $x \in \emptyset$ is never true and then absolutely anything follows (Boyd, 1991, p. 22). Each empty function is thus a *constant function* that produces a constant value c that is defined as $f(x) = c$, for each $x \in X$.

The set of all functions from the domain X to the codomain Y is known as the *function set* and it is denoted by Y^X. This generalization can, however, be limited to the set of two logical or "truth" values, which is 1 for true and 0 for false, i.e. $\mathbf{V} = 0, 1$. Consequently for a subset of X, this set corresponds to the *characteristic* or *indicator function* for $A \subseteq X$ that indicates the membership of an element $\chi_A : X \rightarrow \mathbf{V}$ provided that:

$$\chi_A(x) = \begin{cases} 1, & \text{when } x \in A, \\ 0, & \text{when } x \notin A. \end{cases}$$

As a result of this, 2^X corresponds to the collection of all subsets of the set X, and the set of all binary relations or ordered pairs on X will be $2^{X \times X}$, this latter expression is also denoted by $\mathbf{R}_X$.

An important fact is that in a social system with multiple relations, it can be the case that two different ties whether they are compound or not coincide on the same ordered pairs. So, in order to keep them separate with different labels, it is necessary that the set of ordered pairs $\mathbf{R}_X$ is indexed by the alphabet set $\mathbf{\Sigma}$, which means that the function set $(2^{X \times X})^{\mathbf{\Sigma}}$ will represent a family of relations indexed by $\mathbf{\Sigma}$ of relational labels.

1.9.2 Transformations

Transformations are special types of function, which is an action on a set of elements to itself. If the arrangement of the object set remains invariant after the transformation, then the function represents "symmetry". Symmetry is closely related to group structure and group theory where are two types of transformations involved, namely *rotations* and *reflections*.

These two kinds of transformations, rotations and reflections, play an important role in certain types of permutation groups that are algebraic structure made of the set of all permutations on a given set. We look at some special types of permutation groups in Chapter 2, and in Chapter 3 a group structure representing a human social system with the use of permutation matrices.

Formally, a *transformation* on a set X is a function $f : X \to X$, and the *composition* of two transformations f and g, defined as $(f \circ g)(x) = f(g(x))$, means "apply f to the result of g." The associativity property applies in the composition of transformation functions, whereas commutativity generally does not apply in this case.

1.10 Homomorphism and Congruence

A ***homomorphism*** h is a structure-preserving mapping between algebraic objects that belongs to the *Universal Algebra*, which is the general theory of classes of algebraic structures; i.e., systems that maintain the operation in the involved algebraic objects.

For two algebraic objects S and T, a homomorphism $h : S \to T$ has h as a function from S to T whenever for each $a, b \in S$,

$$h(ab) = h(a)\ h(b).$$

Since homomorphisms are functions that preserve the structure, the products, which in this case result from juxtaposition, are preserved under the homomorphism and, for the two algebraic objects, the operation on the left-hand side corresponds to S while the one on the right-hand side is in T. If $h(a) = h(b)$, then a and b are said to be in the same *congruence class* and this is denoted as an equivalence relation by $a \equiv b$.

If S and T are semigroups (algebraic objects defined in Chapter 2), the function $h : S \to T$ is a *semigroup homomorphism*; and if h performs a simplification of the given system structure, then the mapping is a *homomorphic reduction*. Consequently, the semigroup obtained T, which is the *homomorphic image* of S, is called as a *quotient semigroup* under the equivalence relation on the semigroup that is stable under the operation, and hence the homomorphic reduction of an algebraic structure is a special kind of homomorphism that is surjective. On the other hand, a homomorphism that is a one-to-one mapping, which is also known as monomorphism, permits one to identify how closely the elements of systems are related to each other.

A homomorphism that is a one-to-one correspondence is called an *isomorphism* when is invertible, and two structures are said to be isomorphic when they are structurally identical, differing only in the labels of their elements. Thus, if h is an isomorphism defined on S, then S

and $h(S)$ are isomorphic and this is written as

$$S \cong h(S).$$

Besides, an isomorphism of a system onto itself is called an *automorphism*, and this means that the function is actually a permutation of the elements of the set; here the mapping acts as an identity function, i.e. $h : S \to S$, where h is an automorphism.

1.10.1 Congruence Relations

An equivalence relation $\equiv$ on a semigroup S is *right compatible* if there exists elements $a, b, c \in S^1$ such that $ac \equiv bc$ holds; similarly, $\equiv$ is *left compatible* if $ca \equiv cb$ holds. Any equivalence relation on S that is both left and right compatible is an equivalence relation on the semigroup S; however, one can relax this condition to be just considering a *right-* or *left congruence* whenever the equivalence relation is either simply right or left compatible (Howie, 1996, p. 22; Wu, 1984, p. 293). A right or left congruence will induces respectively a right- or left homomorphism and thus a *right-* or *left homomorphic images* of the algebraic structure.

In formal terms, a ***congruence relation*** $\equiv_h$ that corresponds to the homomorphism h on a semigroup S is an equivalent relation such that for each $a, b \in S$:

$$(a, b) \in \; \equiv_h \quad \text{and} \quad (h(a), h(b)) \in \; \equiv_h \quad \text{implies} \quad (a\, h(a), b\, h(b)) \in \; \equiv_h$$

that corresponds to the *substitution property* of the relation. Besides, any congruence relation is reflexive, symmetric, and transitive, which are the properties of the equivalence relation (Hartmanis and Stearns, 1966; Pattison, 1993).

Furthermore, the equivalence classes of $\equiv_h$ on S define a partition π on the elements of S where:

$$(a {\equiv_h} b) \in \pi \quad \text{and} \quad (h(a) {\equiv_h} h(b)) \in \pi \quad \text{implies} \quad (ab {\equiv_h} h(a)\; h(b)) \in \pi.$$

Although the congruence relation resembles the equivalence notion, any congruence also preserves the operation between the correspondent classes in the algebraic structure, which is not the case with the ordinary equivalence relation.

Each homomorphism determines a congruence relation, and conversely every congruence relation induces a homomorphism. Therefore, we define particular homomorphisms useful for the analysis of social networks in the following chapters, which will allow us performing the task of unfolding the crucial characteristics of multiplex social networks while keeping its essence in the process.

1.10.2 Kernel of a Homomorphism

Before introducing particular homomorphisms for the analysis of social networks, we define the “kernel of a homomorphism”, which is intimately related to the image of the network relational system. For instance, the kernel of a semigroup S is the minimal ideal of S. Thus in a mapping h from S to T, the kernel of a h is a set K of all elements in S that are carried by h to the neutral element of T.

Formally, if 1 represents the neutral element of T, the *kernel* K of a homomorphism h in S is defined as:

$$K = \{ \ a \in S \mid h(a) = 1 \ \}.$$

That is, the homomorphism of element a produces the identity element of the algebraic structure.

1.11 Structural Analysis with Algebra: Summary

This initial Chapter focused on the introduction of fundamental concepts and network representations such as graphs and matrices. Different sorts of algebraic analyses of social networks benefit from these representation forms and from the algebra of relations introduced with formal definitions.

The equivalence and ordering of relations allowed us to establish a network structure as an ordered system made of classes of actors. As a result, we were able to look at the gradation of elements occupying different levels in the systems. We also looked at different types of properties, operations, and functions that are useful for the modeling of social networks. For instance, the relational composition operation plays a central role in the establishment of relational structures of different types of multiplex networks where relational structures link the network ties.

1.12 Learning Structural Analysis by Doing

1.12.1 Getting Started

```
# install the packages from CRAN
R> install.packages("multiplex", "multigraph")
# or their beta versions from GitHub
R> devtools::install_github("mplex/multiplex", ref = "beta")
R> devtools::install_github("mplex/multigraph", ref = "beta")
```

1.12.2 Matrices

```
# load the "multiplex" package
R> library("multiplex")

# create a network with three nodes
R> net <- transf(c("1, 2", "1, 3", "2, 3"))
# create a multiple network with two types of relations
R> net2 <- zbind(net, transf("3, 1"))
```

```
# adjacency matrix representing 'net'
R> net

  1 2 3
1 0 1 1
2 0 0 1
3 0 0 0
```

```
# symmetrize the adjacency matrix
R> mnplx(net, directed = FALSE)

  1 2 3
1 0 1 1
2 1 0 1
3 1 1 0
```

```
# coerce network 'net2' into a "Signed" class object
R> signed(net2)
```

```
$val
[1] 1  0  -1

$s
   1 2 3
1  0 1 1
2  0 0 1
3 -1 0 0

attr(,"class")
[1] "Signed"
```

1.12.3 Graphs

```
# load the "multigraph" package
R> library("multigraph")

# define two scopes with node / edge / graph characteristics
R> scp <- list(cex = 12, vcol = 1, lwd = 9, ecol = 1, rot = -30)
R> scps <- c(scp, signed = TRUE, bwd = .5, swp = TRUE)
```

```
# Fig. 1.1.a.  plot graph with customized format and (default) circular layout
R> multigraph(net, scope = scp, directed = FALSE)

# Fig. 1.1.b.  same as 1.1.a but as (default) directed graph
R> multigraph(net, scope = scp)

# Fig. 1.1.c.  same as 1.1.b but as signed graph
R> multigraph(signed(net2), scope = scps)
```

```
# Fig. 1.2.a.  the complement of Fig. 1.1.a
R> multigraph(1 - mnplx(net, directed = FALSE), scope = scp)

# Fig. 1.2.b.  the converse of Fig. 1.1.b
R> multigraph(t(net), scope = scp)

# Fig. 1.2.c.  the composition of relations in Fig. 1.1.c
R> multigraph(net2[,,1] %*% net2[,,2], scope = scp, loops = TRUE)
```