1
Generalised Shortest Paths Trees: A Novel Graph Class Applied to Semiotic Networks

Alexander Mehler

1.1
Introduction

In this chapter we introduce a class of tree-like graphs which combines the efficiency of tree-like structures with the expressiveness of general graphs. Our starting point is the notion of a *Generalised Tree* (GT), that is, a graph with a kernel hierarchical skeleton in conjunction with graph inducing peripheral edges [17]. We combine this notion with the theory of *Network Optimisation Problems* (NOP) [58] in order to introduce *Generalised shortest PathS Trees* (GPST) as a subclass of the class of generalised trees. One advantage of this novel class is that it provides a functional semantics of the different types of edges of generalised trees. Another is that it gives naturally rise to combining graph modeling with conceptual spaces [27] and, thus, with cognitive or, more generally, semiotic modeling. The chapter provides three examples in support of this combination.

The graph model presented in this chapter focuses on structure formation in semiotic networks. Its background is the rising interest in network models due to the renaissance of, so to speak, functionalist models of networking in a wide range of scientific disciplines starting from the famous work of Milgram [44] in social psychology and extending into the area of physics [1], quantitative biology [3,23], quantitative sociology [6,64], quantitative linguistics [**?**, 40] and information science [47] to name only a few — see [46], [21] and [39] for surveys of this research in the area of natural sciences and the humanities.

What all these network models have in common is that they start from a remarkably low-level graph model in terms of simple graphs with at most labeled or typed vertices and edges. That is, for a decade or so networks have been explored almost exclusively in terms of simple graphs [46] in some cases with weighted edges [4, 51] together with a partitioning into bipartite models [65]. One exception to this trend is the notion of network motifs [52] which is restricted to the formation of micro level structures. That is, the main focus of research has been on structures on the macro level (cf., e.g. the bow-tie

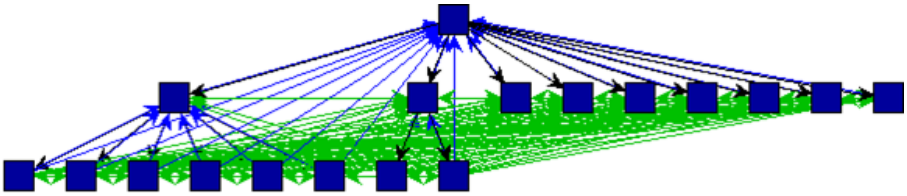

Fig. 1.1: A webgraph [11] in the form of a directed generalised tree derived from a conference website (`www.text-technology.de`) [15, 42].

model of [8]) disregarding intermediary levels of structure formation within complex networks. As a consequence, structure formation is almost exclusively dealt with in terms of network characteristics as a gateway to "universal" laws of network organisation [5]. Note that there are many graph cluster algorithms for identifying subgraphs of an above average cluster-internal homogeneity and cluster-external heterogeneity (see [11] for an overview). This is in the line of supervised or unsupervised learning which basically decide on the membership of objects to clusters as partitions of the vertex sets of the underlying graph. In this chapter we want to shed light on a graph model in the area of semiotic networks which goes beyond traditional approaches to graph clustering and, at the same time, departs from the predominant model-theoretic abstinence regarding meso level structures.

Generally speaking, graph models are quite common in semiotics and related disciplines. Whereas in linguistics tree-like models predominate (as, e.g., rhetorical structure trees [36] to name only one example), there have been made efforts to building more general graph models in quantitative linguistics [7, 37] partly inspired by category theory [28] and topology [26]. To mention only three less cited approaches of this area (see [39] for a survey of network models in linguistics): Firstly, [59] builds, amongst others, on the categorial notion of product and coproduct in order to model the process of meaning constitution in lexical networks. Secondly, [2] utilises hierarchical hypergraphs as models of recursive processes of networking. Thirdly, [22] utilise — comparable with [2] — the notion of a colimit in order to give a formal account of emergent structures in complex systems.

In spite of their expressiveness, category theory and topology are hardly found as methodic bases of present-day approaches in quantitative and computational linguistics. One reason is that graph theory seems to be already expressive enough to master a wide range of structure formations in linguistics. In this chapter we follow this methodic conception, however with a focus on *generalised trees* [16]. In web mining, generalised trees have been introduced in order to grasp the striking gestalt of web documents in-between tree- and graph-like structures [15, 17, 42]. See Figure 1.1 for an example of a

generalised tree with a typical kernel hierarchical structure complemented by graph-inducing lateral and vertical links. Recently, [25] have shown that this concept is also of interest in modeling biological structures. However, one important question has been left open by this research.

Generally speaking, the search for spanning trees of a given graph which satisfy certain topological constraints is a well-known research topic in graph theory [58]. Along this line of research we can make a central question about generalised trees as follows: *Given a connected graph G, which generalised tree G' induced by G satisfies certain desirable topological constraints?* By focusing on this question we do — unlike related approaches — not ask about a similarity model of pairs of predetermined generalised trees (see [16] for such a model). In contrast to this, we take a step back and ask how to induce generalised trees from a given connected graph. This problem is at the core of the present chapter. It will be tackled by means of the notion of a *Generalised Shortest Path Tree* (GSPT). The basic idea behind this notion is to introduce a functional semantics of edge types of generalised trees. That is, starting from a graph we justify in functional terms which of its edges preferably serve as kernel, lateral or vertical links. That way, we introduce a functional semantics into the inducement of generalised trees which goes beyond the approaches mentioned above.

This endeavour is in accordance with [58, p.71] who generally describes the approach of network optimisation as follows: Given a weighted graph $G = (V, E, \mu)$ — called *network* — whose edges are weighted by an edge weighting function $\mu : E \to \mathbb{R}$, the task is to describe a *Network Optimisation Problem* (NOP) which consist of finding a subgraph of G which satisfies a set of well defined properties by optimising (i.e., minimising or maximising) a certain function of μ. It is a basic idea of the present chapter to introduce the notion of context-sensitivity into the specification of such NOPs. That is, as distinguished from the notion of a minimum spanning tree we look for subgraphs in the form of generalised trees whose generation is non-trivially affected by the choice of some root vertex. This sort of context-sensitivity is in accordance with what is known about priming and spreading activation in cognitive networks [45]. As becomes clear by this explanation the present chapter simultaneously aims to provide both a graph-theoretically and empirically well-founded graph model.

What do we gain by such a graph model? Such a model is a first step towards a time and space efficient as well as cognitively plausible model of information processing in semiotic networks. Although we aim at this model, the present chapter does not cut the Gordian knot. That is, our graph model of structure formation in complex networks does not overcome the disregard mentioned at the beginning. What we do is to develop further generalised trees as models of discourse structures. So the main result of this chapter is a formalisation of

a promising concept which may help to better understand structure formation in semiotic networks and the processes operating thereon.

The chapter is organised as follows: In Section 1.2, our graph model is developed in detail, including directed and undirected graphs. At the core of this chapter there is the notion of a generalised shortest path tree which enables a detailed semantics of the different edge types provided by generalised trees. It turns out that this is a step towards combining generalised trees with the theory of conceptual spaces. This combination is also provided by Section 1.2. Next, Section 1.3 gives an empirical account of our graph model by example of three semiotic systems: social tagging, text networks, and discourse structures.

1.2
A Class of Tree-Like Graphs and some of its Derivatives

1.2.1
Preliminary Notions

In this section we briefly define two well-known notions which will be used throughout the chapter to introduce our graph model. This relates to paths in undirected and directed graphs as well as to the notion of geodesic distance in weighted graphs:

Definition 1.1 (Preliminaries) Let $G = (V, E, \mathcal{L}_V, \mathcal{L}_E, \mu)$ be a connected weighted undirected graph whose vertices are uniquely labeled by the function $\mathcal{L}_V \colon V \to L_V$ for the set of vertex labels L_V and whose edges are uniquely labeled by $\mathcal{L}_E \colon E \to L_E$ for the set of edge labels L_E. Throughout this chapter we assume that $\mathcal{L}_V \subset \mathbb{N}_0$ and $\mathcal{L}_E \subset \mathbb{N}_0$, that is, vertices and edges are labeled by ordinal numbers. Further, we assume that this numbering is consecutive. Next, let $D = (V, A, \mathcal{L}_V, \mathcal{L}_E, \nu)$ be an orientation of G, that is, a connected weighted digraph such that $\forall a \in A \colon \nu(a) = \mu(e) \Leftrightarrow e = \{in(a), out(a)\} \in E$. By $\leq_V \subset L_V^2$ ($\leq_E \subset L_E^2$) we refer to the natural order of $L_V \subset \mathbb{N}_0$ ($L_E \subset \mathbb{N}_0$) such that for all $a, b \in V, a \neq b$ ($e, f \in E, e \neq f$): $a <_V b$ ($e <_E f$) iff $\mathcal{L}_V(a) < \mathcal{L}_V(b)$ ($\mathcal{L}_E(e) < \mathcal{L}_E(f)$). This allows us to define the order relation $\leq_a = \leq_V \cup \leq_E$ on the set of vertices *and* edges. Without loss of generality we assume that $\mu : E \to \mathbb{R}^+ \setminus \{0\}$ is an edge weighting function which represents costs of traversing edges in E. Think of μ, e.g., as a function of the loss of coherence induced by following hyperlinks. Analogously, we assume that $\nu(a), a \in A$, represents the cost of entering $out(a)$ when coming from $in(a)$. Now let $\mathbb{P}(G)$ be the set of all simple paths in G and $P = (v_{i_0}, e_{j_1}, v_{i_1}, \ldots, v_{i_{m-1}}, e_{j_m}, v_{i_m}) \in \mathbb{P}(G)$ such that $\forall 1 \leq k \leq m \colon e_{j_k} = \{v_{i_{k-1}}, v_{i_k}\} \in E$. Further, let $\mathbb{P}(D)$ be the set of all simple paths in D and

$\vec{P} = (v_{i_0}, a_{j_1}, v_{i_1}, \ldots, v_{i_{m-1}}, a_{j_m}, v_{i_m}) \in \mathbb{P}(D)$ such that $\forall a_{j_k} \in \{a_{j_1}, \ldots, a_{j_m}\}$: $in(a_{j_k}) = v_{i_{k-1}} \wedge out(a_{j_k}) = v_{i_k}$. Then, $V(P) = \{v_{i_0}, v_{i_1}, \ldots, v_{i_{m-1}}, v_{i_m}\} \subseteq V$ is the set of all vertices, $E(P) = \{e_{j_1}, \ldots, e_{j_m}\} \subseteq E$ the set of all edges and $VE(P) = V(P) \cup E(P)$ the set of all constituents of P. Analogously, we define the sets $V(\vec{P})$, $E(\vec{P})$ and $VE(\vec{P})$ for directed paths $\vec{P}$. If G (resp. D) is a (directed) tree then for each $v, w \in V$ the simple path ending at v and w is unique. Such paths will be denoted as P_{vw} ($\vec{P}_{vw}$) indexed by their end vertices v and w. Next, we define the order relation $\leqslant_a \subseteq \mathbb{P}(G)^2$ on the set of paths $\mathbb{P}(G)$ of G such that for $P = (v_{i_1}, e_{i_2}, \ldots, e_{i_{m_i-1}}, v_{i_{m_i}})$, $P' = (v_{j_1}, e_{j_2}, \ldots, e_{j_{m_j-1}}, v_{j_{m_j}}) \in \mathbb{P}(G)$, $P \neq P'$: $P \leqslant_a P'$ iff $\exists r < \min(m_i, m_j) \forall k \in \{1, \ldots, r\} : VE(P) \ni x_{i_k} = x_{j_k} \in VE(P') \wedge VE(P) \ni x_{i_{r+1}} <_a x_{j_{r+1}} \in VE(P')$. Analogously, we define the order relation $\preccurlyeq_a \subseteq \mathbb{P}(D)^2$ on the set $\mathbb{P}(D)$ of directed paths of D. Further, by $\mathbb{P}_G(v, w)$ we denote the set of all simple paths in G ending at v and w. Finally, for $v_{i_{m_i}} = v_{j_1}$ we define the concatenation $P \circ P' = (v_{i_1}, e_{i_2}, \ldots, e_{i_{m_i-1}}, v_{i_{m_i}}, e_{j_2}, \ldots, e_{j_{m_j-1}}, v_{j_{m_j}})$ of P and P'. $\diamond$

Remark. Throughout this chapter we will always assume the existence of the labeling functions $\mathcal{L}_V$ and $\mathcal{L}_E$ and, thus, of the order relations $\leqslant_a$ and $\preccurlyeq_a$ without explicitly noting this in the subsequent definitions of graphs. The reason of this omission is to keep the formalism simple.

Remark. Why so much effort in defining order relations over paths? The reason is that in semiotic systems, multi- and pseudographs are common (e.g., due to redundancy in the system) while simple graphs seem to be the exception. Think, for example, of graphs as simple as webgraphs in which vertices denote pages while edges stand for hyperlinks. Here, it is not unusual that two pages are linked by different edges distinguished by the location of their anchors within the source page. Using some measure of lexical similarity of interlinked texts [34] such links may be equally weighted. As a consequence, we need a method of distinguishing such edges and the paths built out of them in order to provide uniqueness of the mathematical notions to be defined. This is provided by $\leq_E$ which may explore, for example, the aforementioned positional information.

Based on Definition 1.1 we can now define the notion of a geodesic path:

Definition 1.2 (Geodesic Distance and Geodesic Path) Let $G = (V, E, \mathcal{L}_V, \mathcal{L}_E, \mu)$ be a weighted connected graph according to Definition 1.1. Then, we extend μ as a function of $\mathbb{P}(G)$, that is,

$$\mu : \mathbb{P}(G) \to (0, \infty)$$

such that for each $P = (v_{i_0}, e_{j_1}, v_{i_1}, \ldots, v_{i_{m-1}}, e_{j_m}, v_{i_m}) \in \mathbb{P}(G)$ we set

$$\mu(P) = \sum_{k=1}^{m} \mu(e_{j_k})$$

Based on μ we define the *geodesic path* $GP_\mu(v, w)$ *between* v *and* w *in* G as

$$GP_\mu(v, w) = \inf_{\leqslant_a} \left\{ \arg\min_{P \in \mathbb{P}_G(v,w)} \mu(P) \right\}$$

where $\mathbb{P}_G(v, w)$ is the set of all simple paths in G ending at v and w (see Definition 1.1). Further, the *geodesic distance* $\hat{\mu} : V \times V \to [0, \infty)$ between $v, w \in V$ is defined as

$$\hat{\mu}(v, w) = \begin{cases} 0 & v = w \\ \mu(GP_\mu(v, w)) & v \neq w \end{cases}$$

Finally, for any weighted graph $G = (V, E, \mu)$ we define

$$\mu(G) = \sum_{e \in E} \mu(e)$$

◇

Note that the definition of geodesic distances and paths make use of the order relation $\leqslant_a$ on the set of paths. These two notions play a crucial role in defining so called generalised shortest path trees in Section 1.2.4 and 1.2.5. Further, they are used to bridge the gap between the graph model introduced here and the cognitive-linguistic notion of a conceptual space [27]. This is done in Section 1.2.8 and 1.2.9. We are now in a position to introduce the fundamental notion of a generalised tree.

1.2.2 Generalised Trees

What is common to many semiotic networks is their hierarchical skeleton in conjunction with graph inducing links. Obviously, such networks lie in-between tree-like structures on the one hand and more general graphs on the other. This ambivalent nature has been grasped by the notion of a *Generalised Tree* (GT) [17] which will be developed further in the subsequent sections. Extending the approach presented in [17] and [40] we will distinguish directed from undirected GTs. The reason for doing this is dictated by the nature of semiotic structures: there are semiotic systems which are better described by abstracting from the orientation of arcs used to model relations among their components. This holds, for example, for lexical networks whose nodes are in-

terlinked by multiple arcs or simply by edges. In order to capture the variety of semiotic structures which are spanned over their kernel tree-like skeletons we utilise and extend the following graph-theoretical apparatus:

Definition 1.3 (Undirected Generalised Tree) Let $T = (V, E', r)$ be an undirected tree rooted in r. Let further $P_{rv} = (v_{i_0}, e_{j_1}, v_{i_1}, \ldots, v_{i_{n-1}}, e_{j_n}, v_{i_n}), v_{i_0} = r, v_{i_n} = v, e_{j_k} = \{v_{i_{k-1}}, v_{i_k}\} \in E', 1 \leq k \leq n$, be the unique path in T from r to $v \in V$ and $V(P_{rv}) = \{v_{i_0}, \ldots, v_{i_n}\}$ the set of all vertices of P_{rv}. An *Undirected Generalised Tree* (GT)

$$G = (V, E, \tau, r)$$

induced by T is a pseudograph (i.e. a multigraph which may contain loops) *rooted* in r whose edges are typed by the function $\tau : E \to \mathcal{T} = \{\mathrm{k}, \mathrm{l}, \mathrm{r}, \mathrm{v}\}$ as follows — note that edges $e \in E$ are multisets of exactly two elements which in the case of reflexive edges contain the same element twice:[1]

$$\forall e \in E: \begin{cases} \tau(e) = \mathrm{k} & \Rightarrow & e \in E_{\mathrm{k}} = E' \\ & & \textit{(kernel edges)} \\ \tau(e) = \mathrm{v} & \Rightarrow & e \in E_{\mathrm{v}} = \{\{v, w\} \,|\, v \in V(P_{rw}) \vee w \in V(P_{rv})\} \\ & & \textit{(vertical edges)} \\ \tau(e) = \mathrm{r} & \Rightarrow & e \in E_{\mathrm{r}} = \{\{v, v\} \,|\, v \in V\} \\ & & \textit{(reflexive edges)} \\ \tau(e) = \mathrm{l} & \Rightarrow & e \in E_{\mathrm{l}} = [V]^2 \setminus (E_{\mathrm{k}} \cup E_{\mathrm{v}} \cup E_{\mathrm{r}}) \\ & & \textit{(lateral edges)} \end{cases}$$

such that $E^\tau_{[1]} = \{e \in E \,|\, \tau(e) = \mathrm{k}\}, E^\tau_{[2]} = \{e \in E \,|\, \tau(e) = \mathrm{v}\}, E^\tau_{[3]} = \{e \in E \,|\, \tau(e) = \mathrm{r}\}, E^\tau_{[4]} = \{e \in E \,|\, \tau(e) = \mathrm{l}\}$ where $E = \cup_{i=1}^4 E^\tau_{[i]}$. Because of the interdependence of τ and the sequence of sets $E^\tau_{[i]}, 1 \leq i \leq 4$, we alternatively denote G by $(V, E^\tau_{[1..4]}, r)$ where $e \in E^\tau_{[1..4]}$ iff $e \in \cup_{i=1}^4 E^\tau_{[i]}$. In other words, generalised trees G are interchangeably noted as (V, E, τ, r) and $(V, E^\tau_{[1..4]}, r)$. We say that G is *generalised by its lateral, reflexive and vertical edges*. Further, r is called the *root (vertex)* of G. The generalised tree $G = (V, E, \tau, r)$ induces the undirected tree $\mathrm{kern}(G) = (V, E^\tau_{[1]}, r) = T$ called *kernel (tree)* or *skeleton* of G. Further, the graph $\mathrm{periphery}(G) = (V, \cup_{i=2}^4 E^\tau_{[i]})$ is called *periphery* or *complementary graph* of G. Edges belonging to $\mathrm{periphery}(G)$ are called *peripheral edges* (complementing the set of kernel edges). Finally, the generalised tree (V, E, τ, r, μ) with the edge weighting function $\mu : E \to \mathbb{R}$ is called a *weighted undirected generalised tree*. ◇

1) For a multiset $X = (Y, m), m : Y \to \mathbb{N}_{\geq 1}$, we use the notation $X = \{\underbrace{x, \ldots, x}_{n \text{ times}} \,|\, x \in Y \wedge m(x) = n\}$. Further, $[X]^k$ denotes the set of all subsets of k elements of X [19].

Remark. The reason why we use the implication instead of the equivalence relation in defining the edge typing function τ is that there may be multiple edges which are typed differently. Note further that since GTs are multigraphs the sets $E^{\tau}_{[i]}$, $1 \leq i \leq 4$, do not form a partition of E in the strict sense.

Remark. In order to prevent negative cycles [58, 85] we henceforth assume that μ is a function from E to $\mathbb{R}^+ \setminus \{0\}$. Note that it does not make sense to have zero valued edges, that is, edges e for which $\mu(e) = 0$. Such edges simply do not exist. Further, throughout this chapter we only deal with finite graphs.

Example *Let Graph A in Figure 1.2 be given as a starting point. In this case, we can derive a generalised tree* $C = (V, E^{\tau}_{[1..4]}, 0)$ *from A such that* $E^{\tau}_{[1]} = \{\{0,1\}, \{0,4\}, \{0,6\}, \{1,3\}, \{4,5\}, \{2,5\}\}$, $E^{\tau}_{[2]} = E^{\tau}_{[3]} = \emptyset$ *and* $E^{\tau}_{[4]} = \{\{1,2\}, \{1,6\}, \{2,3\}\}$. *Graph D in Figure 1.2 exemplifies another GT of A also rooted in* 0 *but with a different kernel tree. Finally, Graph E in Figure 1.2 shows a GT rooted in vertex* 4 *thereby exemplifying a vertical edge ending at* 0 *and* 6.

Remark. Unlike [24] but in accordance with [17] we do not define GTs by means of a multi-level function. We rather focus, more generally, on *unleveled* graphs. The reason is that in semiotic systems we hardly observe such a mapping of vertices onto distinguished levels of a graph. Look, for example, on the category graph of the Wikipedia [62]: other than possibly claimed by its users, the categories in this graph do not span a tree, but a directed cyclic graph. This fact is contrary to mapping vertices to the same level of taxonomic resolution even if they have the same geodesic distance to the root of the category system which, by the way, does not uniquely exist in this example of social tagging. Further, as we do not only focus on categorical systems with a hierarchical skeleton but additionally on association networks, the idea of a level function gets obsolete. Thus, we need a more general definition of generalised trees as provided by Definition 1.3.

In the sequel of this chapter, the notion of a generalised sub-tree of a GT will be used. By exploring the type system of generalised trees such sub-trees are defined as follows:

Definition 1.4 (Type Restricted Generalised Sub-Tree) Let $G = (V, E, \tau, r, \mu)$ be a weighted generalised tree. Let further $r \in V$ be a vertex in G. Then, for a subset $\mathcal{T}' = \{\mathrm{k}, \ldots\} \subseteq \mathcal{T} \leftarrow E : \tau$ and the restriction τ' of τ to $\mathcal{T}'$, a *Type Restricted Generalised Sub-Tree* of G is a generalised tree $G' = (V, E', \tau', r, \mu')$ of G such that $\forall e \in E\colon \tau(e) \notin \mathcal{T}' \Rightarrow e \notin E' \subseteq E$. In cases where $\mathcal{T}'$ omits types in descending order of the index of the sequence $E^{\tau}_{[i]}$, $1 < i$, we alternatively denote type restricted generalised sub-trees by $(V, E^{\tau'}_{[1..i]}, r, \mu')$ for $i > 1$. As usually, μ' is the restriction of μ to E'. $\diamond$

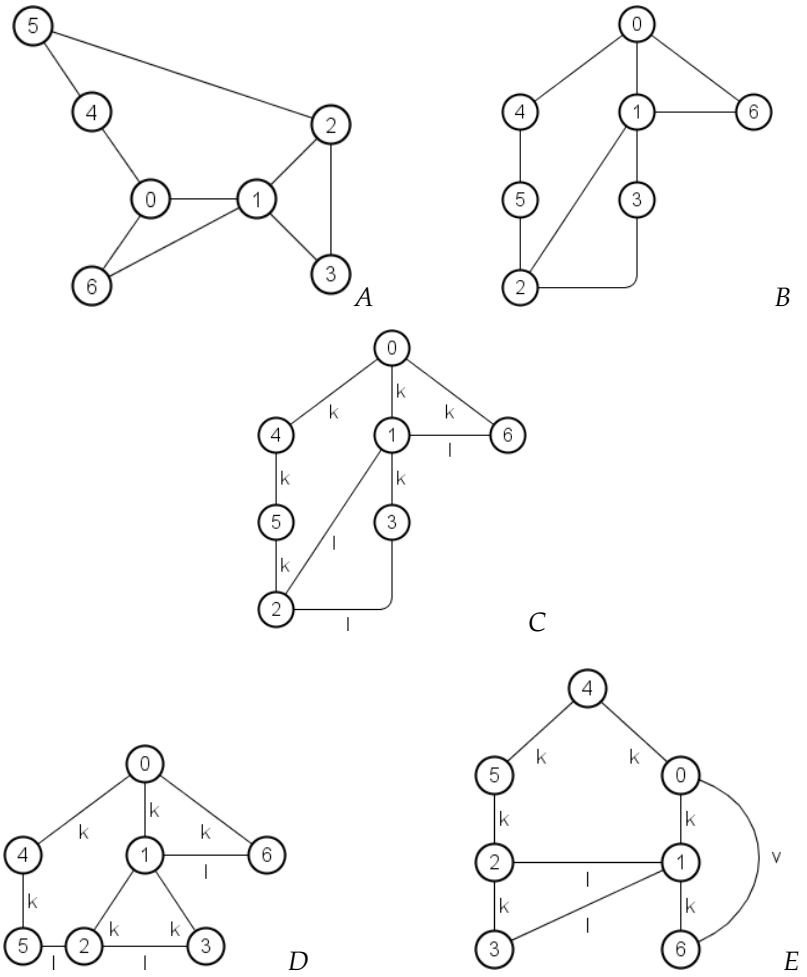

Fig. 1.2: A connected graph A. The same graph in a tree-like perspective denoted by graph B. A generalised tree C of A rooted in 0. A generalised tree D of A rooted in 0 with a different kernel than C. Finally, a generalised tree E of A rooted in 4. For reasons of simplification, edge weights are omitted while edge types are noted as edge labels.

Remark. As generalised trees are connected graphs we do not consider type restricted sub-trees which exclude kernel edges. Therefore, we always have that $\mathrm{k} \in \mathcal{T}'$.

It seems natural to map the tree-like structure of a semiotic network by means of the kernel edges of a corresponding generalised tree G while its peripheral edges may be used to map the remainder of that network. However, as the same graph induces several generalised trees (see, e.g., Figure 1.2) we have to pose the following question: *Given an undirected connected graph*

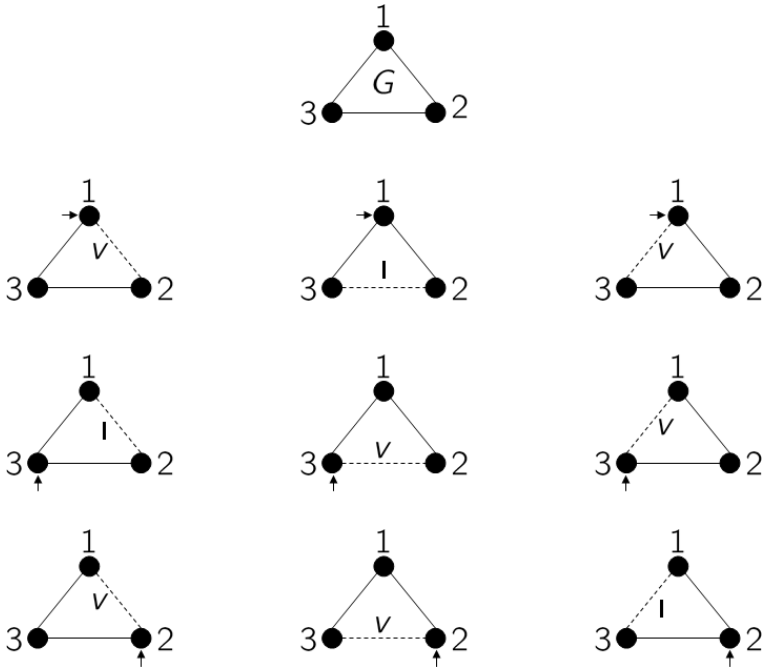

Fig. 1.3: A graph G of three vertices together with all generalised trees derivable from it. Vertical edges are labeled by v, lateral edges by l. Roots are marked by an incoming arrow $\rightarrow$.

$G = (V, E, \mu)$, *how many generalised trees can we built out of* G*?* Look, for example, at Figure 1.3: Given the graph G with three vertices we can derive exactly nine GTs from G in which the third edge is either a vertical edge (if ending at the corresponding root) or a lateral edge. More generally, if G is a completely connected graph with n vertices, there exist n^{n-1} generalised trees of G. The reason is that the cardinality of this set of GTs equals the number of vertices in G times the cardinality of the set of spanning trees of G. And the latter cardinality is determined by n^{n-2} [67]. *But why do both of these sets have the same size?* The reason is that a generalised tree according to Definition 1.3 is determined by its kernel (spanning) tree — remember that apart from kernel edges all other edge types are defined with respect to the kernel tree.

In order to stress the relationship of a graph $G = (V, E, \mu)$ with its generalised trees we add the following definition:

Definition 1.5 (Generalised Spanning Tree) Let $G = (V, E, \mu)$ be a weighted connected undirected graph without negative cycles. Let further $r \in V$ be a vertex in G. An *undirected Generalised Spanning Tree* (GST) $G' = (V, E^{\tau}_{[1..4]}, r, \mu)$ of G is a generalised tree with the kernel $\mathrm{kern}(G') = (V, E^{\tau}_{[1]}, r, \nu)$ as a spanning tree $T = \mathrm{kern}(G')$ of G such that $E = \cup_{i=1}^{4} E^{\tau}_{[i]}$ and ν is the restriction of μ

to $E^{\tau}_{[1]}$. We say that G' *is spanned over* G *by means of* T *starting from* r. G is called the *underlying graph* of G'. ◇

From this perspective, a GT simply denotes a sort of "partitioning" of the set of edges of its underlying graph G: it neither contains more nor less, but exactly the same number of edges as G. The sole, however informative exception is that edges in GTs are *typed* according to the structural classes distinguished by Definition 1.3. What these types actually denote depends on the application area in which GTs are empirically observed. Moreover, a GT is determined by the choice of a spanning tree of the underlying graph: once this kernel is specified together with the root of the GT to be built, its peripheral edges are uniquely determined. Therefore, as long as we do not select a subset of edges but retain the complete edge set of an input graph when deriving a generalised tree from it, we have to focus on the choice of the kernel tree in order to specify subclasses of the class of generalised trees. From a formal point of view, this is the central theme of this chapter: We ask about the impact of this choice on typing peripheral edges and shed light on their semantics as a result of this choice. This semantics is in turn our bridge to empirical systems which because of their structural constraints impose different semantics of kernel and peripheral edges. This will be specified in the remainder of the chapter.

Before we proceed introducing subclasses of the class of generalised trees we first ask about the time complexity of computing the sort of edge typing induced by them. This is answered by the proof of the following theorem:

Theorem 1.1 *Given a connected graph* $G = (V, E, \mu)$ *without negative cycles, a vertex* $r \in V$ *and a spanning tree* $T = (V, E', \nu)$ *of* G*, the time complexity of computing the generalised spanning tree* $G' = (V, E^{\tau}_{1..4}, r, \mu)$ *spanned over* G *by means of* T *starting from* r *is in the order of* $\mathcal{O}(|V| + |E|)$.

Proof. First, we observe that solving this task basically demands distinguishing between vertical and lateral edges. The reason is that while kernel edges are identified by their membership to T, reflexive edges are distinguished by the fact that they contain the same vertex twice. Because of the definition of lateral edges this further means that we have to decide whether a given edge $e \in (E^{\tau}_{[1..4]} \setminus E^{\tau}_{[1]}) \setminus E^{\tau}_{[3]}$ is a vertical edge. This decision is computed by Algorithm 1 whose time complexity can be estimated as follows:

- Line 3 computes a vector of all paths starting from r and ending at some vertex $v \in V$. We suppose that all vertices are indexed consecutively (by the labeling function $\mathcal{L}_V$ — see Definition 1.1) so that any path P_{rv} can be accessed in $\mathbf{x}$ by v's index. That way, generating $\mathbf{x}$ can be carried out by a breadth first search of order $\mathcal{O}(|V| + |E|) = \mathcal{O}(|V| + |V| - 1) = \mathcal{O}(|V|)$.
- Line 8 denotes an index-based access operation which in the case of a vector is of constant complexity [57].

Algorithm 1 Spanning Peripheral Edges

Require: A graph $G = (V, E, \mu)$, a spanning tree $T = (V, E', \nu)$ of G and a vertex $r \in V$ according to Definition 1.5.

Ensure: The set $E^{\tau}_{[2]}$ of vertical, the set $E^{\tau}_{[3]}$ of reflexive and the set $E^{\tau}_{[4]}$ of lateral edges of the GST G' spanned over G by means of T starting from r.

```
1:  procedure SpanningPeripheralEdges(G, T, r)
2:      E^τ_[1] ← E'; E^τ_[2] ← E^τ_[3] ← E^τ_[4] ← ∅
3:      x ← VectorOfAllPathsInTreeStartingFromRoot(T, r)
4:      for e = {v, w} ∈ E \ E^τ_[1] do
5:          if v = w then
6:              E^τ_[3] ← E^τ_[3] ∪ {e}
7:          else
8:              v ← x[v] ∧ w ← x[w]
9:              if v[w] ∨ w[v] then
10:                 E^τ_[2] ← E^τ_[2] ∪ {e}
11:             else
12:                 E^τ_[4] ← E^τ_[4] ∪ {e}
13:             end if
14:         end if
15:     end for
16:     return E^τ_[2..4]
17: end procedure
```

- Line 9 denotes a search operation which is also of constant complexity if paths are represented as vectors of length $|V|$. That is, $\mathbf{x}[v]$ is a Boolean vector such that for any $w \in V$: $\mathbf{x}[v][w] = 1 \Leftrightarrow w \in P_{rv}$ — note that we only check for membership of vertices in paths.
- Line 4 requires the repetition of Line 5–14 exactly $|E| - |E'|$ times which because of the constant complexity of the latter operations is in the order of $\mathcal{O}(|E| - |E'|) = \mathcal{O}(|E|)$.

Thus we get $\mathcal{O}(|V| + |E|)$ as the desired upper bound. Of course, more efficient algorithms can be designed but are not of interest in this chapter as Algorithm 1 is already sufficiently efficient. □

Remark. Utilising Algorithm 1 the computation of generalised spanning trees is divided into two parts: firstly, computing the kernel spanning tree T and, secondly, typing the remainder of lateral, reflexive and vertical edges. Below we will reuse this greedy approach in order to estimate the time complexity of generating generalised trees whose kernel trees meet specific constraints.

1.2.3 Minimum Spanning Generalised Trees

Based on the notion of a generalised spanning tree and on the fact that the number of these generalised trees derivable form a connected graph G is a simple function of the number of its spanning trees we can pose a more interesting question: *Which GTs among all possible spanning GTs of a connected graph G meet which structural constraints?* This question goes beyond a purely mathematical endeavour as it bridges the area of empirical, i.e. semiotic systems on the one hand and mathematical systems on the other. The reason is that interesting structural constraints are those for which there are relevant semiotic or information-theoretic interpretations. From this point of view not all but only a subset of generalised trees is worth being considered theoretically thereby stigmatising the remainder of generalised trees as semiotically irrelevant. A subclass of generalised trees of a graph G which is certainly of higher relevance due to its impact on the flow of information within G is the one whose kernel tree is spanned by the *Minimum Spanning Tree* (MST) [58] of G.[2] This notion relates to the class of GTs G' of a graph $G = (V, E, \mu)$ whose kernel tree minimises the cost of edge transitions among all candidate spanning trees of G while every peripheral edge ending at some vertices $v, w \in V$ is at least as costly as the kernel edges of G' ending at least at one of these vertices. This notion is captured by the following definition:

Definition 1.6 (Minimum Spanning Generalised Tree) Let $G = (V, E, \mu)$ be a weighted connected graph, $T = (V, E', \nu)$ a minimum spanning tree of G and $r \in V$. The *Minimum Spanning Generalised Tree* (MSGT) induced by T is a generalised tree $G' = (V, E^{\tau}_{[1..4]}, r, \mu)$ spanned over G by means of the kernel tree T starting from r. ◇

Corollary 1.1 *Given a graph $G = (V, E, \mu)$ according to Definition 1.6 and a vertex $r \in V$, the MSGT spanned over G starting from r is not necessarily unique.*

This property is a simple consequence of the fact that already the MST of a graph is not necessarily unique, especially if we consider equally weighted multiple edges. In order to secure uniqueness in this case one can proceed as follows:

Definition 1.7 (Minimum Spanning Generalised Tree Revisited) Let $\mathrm{mst}(G)$ be the set of all minimum spanning trees of the weighted connected labeled graph $G = (V, E, \mathcal{L}_V, \mathcal{L}_E, \mu)$ according to Definition 1.1. Then, we define the

2) Note that the MST of a connected graph G is not necessarily unique. However, utilising the order relation $\leq_a$ of vertices and edges (see Definition 1.1) we can uniquely determine one of these equally weighted spanning trees of minimum weight. This approach can be followed whenever uniqueness is desired with respect to the mathematical constructs introduced subsequently. For an example see Definition 1.7 (below).

order relation $\leq_{\text{kern}} \subseteq \text{mst}(G)^2$ such that $\forall T' = (V, E', \nu'), T'' = (V, E'', \nu'') \in \text{mst}(G) : T' \leq_{\text{kern}} T'' \Leftrightarrow T' = T'' \vee \sum_{e \in E'} \mathcal{L}_E(e) < \sum_{e \in E''} \mathcal{L}_E(e)$. For a given vertex $r \in V$, the $\leq_{\text{kern}}$-*induced Minimum Spanning Generalised Tree* of G is a generalised tree $G' = (V, E^{\tau}_{[1..4]}, r, \mu)$ spanned over G by means of $\inf_{\leq_{\text{kern}}} \text{mst}(G)$ starting from r. ◇

This definition shows a way to derive uniquely defined MSGTs wherever needed. As mentioned above, the derivation of a generalised tree from a graph G generally includes two steps: firstly, generating the spanning tree with its kernel edges and, secondly, typing the rest of edges of G. Following this approach, the time complexity of generating an MSGT is bound by the sum of the complexity of generating its kernel tree and that of typing its peripheral edges. This idea is reflected by the following corollary — note that typing peripheral edges may be done simultaneously with spanning kernel edges so that there are certainly lower complexity bounds than the one mentioned in the following corollary.

Corollary 1.2 *Because of Theorem 1.1 the time complexity of generating an MSGT is in the order of* $\mathcal{O}(|V| + |E| + |E| \log |V|)$ *when using a standard algorithm [58] to generate the kernel MST of the MSGT. It reduces to* $\mathcal{O}(|V| + |E| + |E|\alpha(|E|, |V|))$ *when using the algorithm of [12] where* α *is the classical functional inverse of Ackermann's function.*

The following theorem presents a first important statement about the semantics of peripheral edges — in this case as the result of using a MST as the kernel of a GT:

Theorem 1.2 *Let* $G' = (V, E^{\tau}_{[1..4]}, r, \mu)$ *be an MSGT spanned over* $G = (V, E, \mu)$ *by means of the minimum spanning tree* $T = (V, E', \nu)$ *starting from* r*. Then,* $\forall e \in E^{\tau}_{[1..4]} \setminus E' \forall f \in E' : e \cap f \neq \emptyset \Rightarrow \mu(f) \leq \mu(e)$.

Proof. Let $e = \{v, w\} \in E^{\tau}_{[1..4]} \setminus E'$. Let further $P_{rv} = (r, \ldots, v', f, v)$ and $P_{rw} = (r, \ldots, w', g, w)$ be the unique paths in T from r to $v \in V$ and $w \in V$, respectively. Then, we have to distinguish two cases:

- *Case A (vertical edges):* w is a vertex on P_{rv} or v is a vertex on P_{rw}. In this case we conclude as follows: Without loss of generality we assume that v is a vertex on P_{rw}. Then, we can construct a tree $T' = (V, (E' \setminus \{g\}) \cup \{e\})$ such that $\mu(T') < \mu(T)$ — T and T' differ by a single edge. This contradicts the status of T as a MST of G. Note that changing f by e would disconnect T in the present case where we assume that $v \in P_{rw}$.
- *Case B (lateral edges):* Neither is w a vertex on P_{rv} nor v a vertex on P_{rw}. That is, v and w belong to different branches of T. In this case we conclude as follows: If $\mu(e) < \mu(f)$ we can construct a tree $T' = (V, (E' \setminus \{f\}) \cup \{e\})$ such that $\mu(T') < \mu(T)$ once more in contrast to

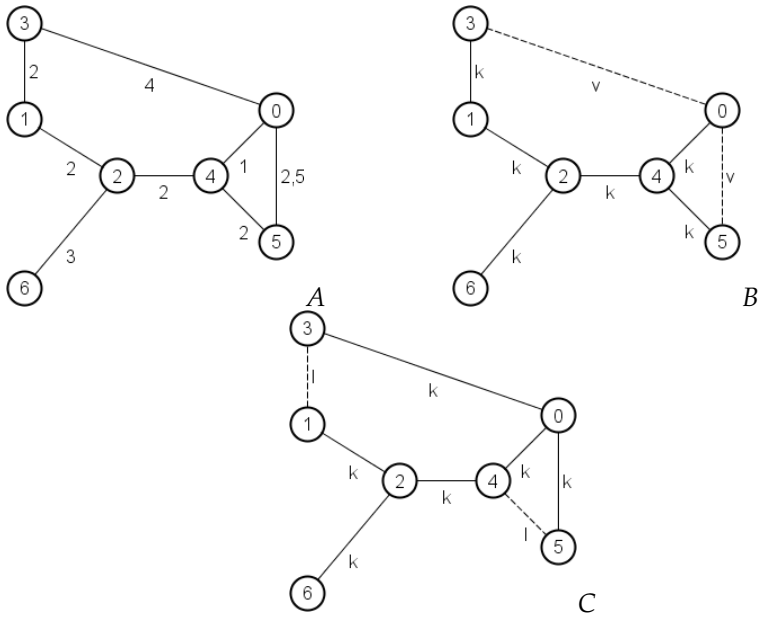

Fig. 1.4: A graph A together with an MSGT (cf. graph B) rooted in vertex 0 and a GSPT (cf. graph C) also rooted in 0 derived from A. For reasons of simplification, edge weights are omitted in the graphical representations of graph B and C.

the status of T as a MST of G. Analogously, we conclude when changing g by e.

□

Remark. Theorem 1.2 separates MSGTs from ordinary graphs G and the rest of GTs derived from G as they distinguish peripheral from kernel edges in terms of μ. In MSGTs, the function of peripheral edges is separated from that of kernel edges: information flow along the former is more costly than along the latter. In spite of this information, MSGTs do not separate vertical from lateral links: Irrespective of its status as a vertical link or as a lateral link, an edge may contribute to paths shorter than that on the kernel minimum spanning tree — note that MSTs are *not* shortest path trees. In this sense, MSGTs do not explore the full informational capacity of generalised trees. Below we will introduce generalised shortest path trees which additionally provide this latter information value.

Example *Let the graph $A = (V, E, \mu)$ be given as shown in Figure 1.4. In this case, we can derive the MSGT $B = (V, E^{\tau}_{[1..4]}, 0, \mu)$ with the kernel minimum spanning tree $T = (V, \{\{0,4\},\{4,5\},\{2,4\},\{1,2\},\{2,6\},\{1,3\}\})$ and the following subsets of edges: $E^{\tau}_{[2]} = \{\{0,5\},\{0,3\}\}$, $E^{\tau}_{[3]} = E^{\tau}_{[4]} = \emptyset$.*

Now we are in a position to repeat the basic principle of generating generalised trees as follows: starting from an underlying graph we select a GT whose kernel and periphery meet certain structural constraints. By specifying these constraints we get more and more informative GTs. This principle is followed in the subsequent sections.

1.2.4 Generalised Shortest Path Trees

It is Theorem 1.2 which motivates speaking about a *minimum* spanning generalised tree. It divides an underlying graph $G = (V, E, \mu)$ into the set of *low cost* (i.e. kernel) and *high cost* (i.e. peripheral) edges. In this sense, MSGTs share the restricted view of a graph G with its MSTs but additionally retain complete information about the topology of G. Note that for different vertices $v, w \in V$ used to root the kernel MST of G, E is separated differently into vertical and lateral edges (while the set of kernel edges remains the same if the MST is uniquely defined). Obviously, this is a very low degree of context-sensitivity induced by the choice of a root which solely affects the typing of edges while leaving the kernel tree untouched. In order to exceed this lower bound of context-sensitivity we can think of GTs whose kernel varies with the choice of the root, that is GTs whose construction is context sensitive to the root being chosen. A candidate instance of this notion is given by generalised shortest path trees which — by analogy to Definition 1.6 — are defined as follows:

Definition 1.8 (Generalised Shortest Path Tree) Let $G = (V, E, \mu)$ be a weighted connected undirected graph without negative cycles, $r \in V$ a vertex and $T_r = (V, E', r, \nu)$ the *Shortest Path Tree* (SPT) of G rooted in r. The *Generalised Shortest Path Tree* (GSPT) induced by T_r is a generalised tree $G' = (V, E^{\tau}_{[1..4]}, r, \mu)$ spanned over G by means of T_r starting from r. ◇

Corollary 1.3 *Given a graph $G = (V, E, \mu)$ according to Definition 1.8 and a vertex $r \in V$, the GSPT induced by T_r is not necessarily unique.*

Once more, this property simply follows from the fact that the underlying graphs of generalised trees may contain equally weighted multiple edges. As before, we can utilise $\leq_E$ to provide uniqueness. By analogy to Corollary 1.2 we get the following corollary:

Corollary 1.4 *According to Theorem 1.1 the time complexity of generating a GSPT is in the order of $\mathcal{O}(|V| + |E| + |V|^2) = \mathcal{O}(|V|^2)$ when using Dijkstra's algorithm [20] to solve the single source problem of computing shortest paths. It reduces to $\mathcal{O}((|V| + |E|) \log |V|) = \mathcal{O}(|E| \log |V|)$ when operating on sparse graphs (for which $|E| \ll |V|^2$).*

Remark. The notion of a generalised shortest path tree is reminiscent of the notion of a distance function-based GT as introduced by [24]. The difference is that [24] use the geodesic distance of vertices from the root of a GT to map

equally distant vertices onto the same level. In contrast to this, we start from a shortest path tree in order to get a kernel tree disregarding any graph levels. It seems that our notion is more general than the one of [24] as it does not refer to structural constraints hardly observable in empirical systems, that is, the questionable existence of graph levels (see above). Nevertheless, both notions pave the way for more complex kernel trees of GTs whose structure is restricted by observable constraints of natural systems (see Section 1.2.8 and 1.2.9).

Example *Let graph A in Figure 1.4 be given as a starting point. Then, graph C in Figure 1.4 is a generalised tree induced by G with the kernel shortest path tree* $T = (V, \{\{0,3\},\{0,4\},\{0,5\},\{2,4\},\{1,2\},\{2,6\}\},0)$ *rooted in* 0 *and the following subsets of edges:* $E^{\tau}_{[2]} = E^{\tau}_{[3]} = \emptyset$, $E^{\tau}_{[4]} = \{\{1,3\},\{4,5\}\}$.

Unlike MSGTs, GSPTs are context-sensitive not only with respect to the classification of peripheral edges, but also with respect to the kernel tree itself which may vary with the choice of the root of the GSPT. That way, we get a one-to-many relation: the same underlying graph $G = (V, E, \mu)$ is non-trivially related to many different kernel shortest path trees (with different edge sets) and, thus, to as many different GSPTs. In the extreme case we get $|V|$ many different kernel trees of GSPTs starting from the underlying graph $G = (V, E, \mu)$. *Non-trivially* means that we disregard multiple edges in this counting (which may induce different GSPTs rooted in the same vertex). In contrast to this, the kernel trees of all MSGTs induced by the various root vertices $v \in V$ have the same edge set as long as the MST of G is unique. Thus, MSTs lack the kind of context-sensitivity of GSPTs and, therefore, do not induce the aforementioned one-to-many relation.

By analogy to Theorem 1.2 we now consider Corollary 1.5 and Theorem 1.3 about GSPTs which together provide a functional separation of vertical and lateral edges in relation to kernel edges (as absent in MSGTs):

Corollary 1.5 *Let* $G' = (V, E^{\tau}_{[1..4]}, r, \mu)$ *be a GSPT spanned over* $G = (V, E, \mu)$ *by means of the SPT* $T_r = (V, E', r)$ *starting from r. Then, for any* $v \in V$ *each path* $P = (r, e_{i_1}, \ldots, e_{i_m}, v)$ *in* G' *ending at r and v is at least as costly as the unique path* P_{rv} *in* T_r*, that is,* $\mu(P) \geq \mu(P_{rv})$.

This corollary is in a sense obvious that we can skip its proof (it is a simple consequence of the definition of SPTs). Its meaning is to assign vertical and lateral edges marginal roles in relation to kernel edges: Starting from the root of a GSPT it is more costly to traverse lateral or vertical edges than following kernel edges. An obvious implication of Corollary 1.5 runs as follows:

Corollary 1.6 *Any vertical edge in a GSPT is at least as costly as the corresponding sub-path of the kernel SPT which is cut short by this vertical edge.*

So far, we distinguished lateral and vertical from kernel edges. This does not really make it beyond the notion of an MSGT. Thus, we have to addition-

ally ask: *How can we further separate the function of lateral edges from the one of vertical edges in GSPTs?* This question is answered by the proof of the following theorem:

Theorem 1.3 *Let $G' = (V, E^{\tau}_{[1..4]}, r, \mu)$ be a GSPT spanned over $G = (V, E, \mu)$ by means of the SPT $T_r = (V, E', r)$ starting from r. Then, the shortest path $GP_{\mu'}(v, w)$ between any pair of vertices v, w in $T' = (V, E^{\tau}_{[1..3]}, r, \mu')$ is a path in T_r — μ' is the restriction of μ to $E^{\tau}_{[1..3]}$. In other words: apart from lateral edges $e \in E^{\tau}_{[4]}$, shortest paths in G' solely contain kernel edges.*

Proof. For $v, w \in V$, P_{rv}, P_{rw} are the shortest paths in T_r ending at r as well as v and w, respectively. Now we have to consider two cases:

- *Case A:* $v \in V(P_{rw})$: In this case we conclude that the sub-path $P = (v, e_{i_1}, \ldots, e_{i_m}, w)$ of the shortest path P_{rw} from r to w is the shortest path between v and w. Otherwise, if there is at least one vertical edge e between two vertices $x, y \in V(P)$ such that $\mu((v, e_{i_1}, \ldots, x, e, y, \ldots, e_{i_m}, w)) < \mu(P)$, then P_{rw} including the sub-path P is not the shortest path between r and w — in contrast to the definition of shortest path trees. $w \in V(P_{rv})$ is a mirror case.
- *Case B:* $v \notin V(P_{rw}) \wedge w \notin V(P_{rv})$, that is, v and w belong to different branches of T_r rooted in r. In this case, there is a unique *least common predecessor* u of v and w in T_r such that $u \in V(P_{rv})$, $u \in V(P_{rw})$ and for any other vertex $x \neq u \in V$ satisfying the same conditions it holds that $x \in P_{ru}$. Now we conclude that $P_1 = (v, e_{i_1}, \ldots, e_{i_m}, u)$ is the shortest path in T_r from v to u and $P_2 = (u, e_{j_1}, \ldots, e_{j_n}, w)$ the shortest path in T_r from u to w. Further, by *Case A* we know that neither P_1 nor P_2 is shortened by including any vertical edge. Thus, $P_1 \circ P_2 = (v, e_{i_1}, \ldots, e_{i_m}, u, e_{i_{j_1}}, \ldots, e_{i_{j_n}}, w) = GP_{\mu'}(v, w)$ in T'.

□

According to this theorem, vertical edges do not shorten any shortest path in the kernel tree of a GSPT. However, things look different if lateral edges are taken into account which may cut short the paths in T_r. For example, in the GSPT in Figure 1.4 (see graph C), the shortest path between vertex 4 and 5 is spanned by a single lateral edge. Together with Theorem 1.3 this observation assigns vertical and lateral edges quite different roles *in GSPTs* so that this class of GTs is more informative about their peripheral edges than MSGTs:

- *Vertical edges* do not shorten any path of the kernel tree of a GSPT. Their role is rather to provide aggregations of such paths at the expense of a more costly transition *of* or less efficient information flow *within* the kernel tree. From the point of view of social taxonomies one can think of vertical edges as condensations in terms of [29]: In a concept hierarchy

induced by hypernymy relations, vertical edges provide short cuts by relating specific with general terms thereby by-passing (i.e. aggregating or condensing) intermediary hypernyms.[3]

- Lateral edges do not have this role in GSPTs. Their function changes between genuine *short cuts* on the one hand — as they enable faster information flow or less costly graph transitions — and *cross references* on the other (which realise more expensive shifts in direction). GSPTs are underspecified with respect to this distinction. Thus, we need a more informative notion of a generalised tree which goes beyond GSPTs — this extension is introduced in Section 1.2.5.

At this point we come back to one of the central themes of the present chapter, that is, the context-sensitive formation of generalised trees spanned over a certain semiotic network. As seen above, MSGTs are less sensitive to the selection of the root of a GT than GSPTs. From a formal point of view, this difference is manifested by Corollary 1.2 in contrast to Corollary 1.5, 1.6 and Theorem 1.3. From an empirical point of view, the choice between MSGTs, GSPTs or even more restricted GTs depends on the characteristics of the natural system under consideration to be retained by its formal model. This can be formulated by means of a criterion as follows: *Whenever we observe a sensitivity of having an overview of a given network subject to adopting an initial position in that network, we have to prefer GSPTs to MSGTs. Otherwise, we can rely on MSGTs which provide invariant kernel trees irrespective of our initial (root) position in the network.* Thus, we can make our question about the preferred model of a generalised tree pointed by asking: *Are there empirical systems in which rooting is decisive at least in the sense as reflected by GSPTs?* This, of course, holds for all cognitive processes based on priming and spreading activation [48, 53, 60]. In this sense, GSPTs are one step towards an empirically well observable concept of structure formation.

1.2.5 Shortest Paths Generalised Trees

In spite of the latter considerations we may complain that unlike MSTs which realise selections of subsets of edges of the corresponding input graph G, MSGTs and GSPTs always include all edges of G. That is, according to Definition 1.3 and 1.5, GTs induce classifications of G's edges and, thus, contain as many edges as G. In order to circumvent this situation we have at least two alternatives:

3) Think, for example, of a situation in which a certain concept, say *small car*, is classified (by a group of interlocutors) nearly as often as a *vehicle* just like as a *car*. In this case, the classification by the more general noun *vehicle* by-passes that by *car* without making the latter obsolete.

- We may start from a kernel MST in order to span the periphery of a GT by the shortest paths between its vertices. This notion preserves information about cheapest edges (to let the resulting GT be efficiently manageable as a tree) *and* about shortest paths (as the cheapest graph-like skeleton of the underlying graph) while it disregards the rest of edges.
- Alternatively, we may generalise the notion of a GSPT by shortest path trees itself. That is, as in the latter case we span the periphery of a GT solely by means of shortest paths, but now around a kernel SPT so that the resulting GT solely consists of shortest path trees — it declares a single SPT as its kernel while the rest of SPTs spans its periphery.

It is the latter notion which is of interest here: it combines the context-sensitivity of GSPTs with a finer grained semantics of peripheral edges, finer than in the case of MSGTs. This combination is grasped by the following definition:

Definition 1.9 (Shortest Paths Generalised Tree) Let $G = (V, E, \mu)$ be a weighted connected undirected graph without negative cycles, $r \in V$ any vertex and $T_r = (V, E', r, \nu)$ the SPT of G rooted in r (as usual, ν is the restriction of μ to E'). The *Shortest Paths Generalised Tree* (SPGT) $G' = (V, E^{\tau}_{[1..4]}, r, \mu')$ derived from G is a GSPT spanned over the simple graph $G'' = (V, E'', \mu')$ by means of T_r starting from r such that

$$E'' = \bigcup_{v \in V, T_v = (V, E_v, v, \mu_v)} E_v$$

E'' is the set of all edges belonging to any SPT of any vertex of G; T_v is the SPT induced by $v \in V$ in G. Finally, μ' is the restriction of μ to $E^{\tau}_{[1..4]}$. ◇

Remark. We assume that G'' is a simple graph and therefore neither contains loops nor multiple edges. Otherwise, the resulting SPGT of a graph would contain *more* edges than its underlying graph (because of the union which is in use in Definition 1.9).

Based on this definition we can easily prove the following corollary:

Corollary 1.7 *Let $G' = (V, E^{\tau'}_{[1..4]}, r, \mu)$ be a GSPT spanned over $G = (V, E, \mu)$ by means of the SPT $T_r = (V, E', r, \nu)$ starting from $r \in V$. Then, the SPGT $G'' = (V, E^{\tau}_{[1..4]}, r, \mu'')$ derived from G by means of T_r and r satisfies the following equalities and one inequality:*

1. $E^{\tau}_{[2]} = \emptyset$
2. $E^{\tau}_{[3]} = \emptyset$
3. $E^{\tau}_{[4]} \subseteq E^{\tau'}_{[4]}$
4. $\forall e = \{v, w\} \in E^{\tau}_{[4]} : \mu(e) \leq \mu(P_{vw})$ — *note that P_{vw} denotes the unique path in the kernel tree of G' ending at v and w (see Definition 1.1).*

Proof. Case 1 is a consequence of Theorem 1.3, Case 2 is a consequence of the fact that shortest paths are always simple while Case 3 and 4 are simple consequences of the way SPGTs are defined, that is, for generating a shortest path between two vertices v, w in G'' a lateral edge $\{x, y\}$ is added to $E^{\tau}_{[4]}$ if and only if it shortens the path P_{xy} as a sub-path of P_{vw}. □

Remark. If we define G'' in Definition 1.9 as a multigraph then the periphery of the SPGT derived from it is always connected. The reason is that in this case all kernel edges are duplicated as often (in the form of peripheral edges) as there are different SPTs of G to which they belong. This characteristic of "peripheral connectivity" is not necessarily provided by any of the concurrent notions of GTs introduced so far. However, such a highly connected GT would contain more edges than its underlying graph. In the present stage of modeling this characteristic is not desirable. It may be the starting point for future extensions of the notion of a generalised tree.

In the next section we utilise the notion of a shortest path generalised tree in order to make the next step in specifying a functional semantics of edges in generalised trees. As will be shown, this is accompanied by an extension of the set of edge types used so far.

1.2.6 Generalised Shortest Paths Trees

Based on the notion of an SPGT we can now define generalised trees in which short cuts are provided by a separate, more specific edge type. This is done by means of the notion of a generalised shortest path**s** tree:

Definition 1.10 (Generalised Shortest Paths Tree) Let $G = (V, E, \mu)$ be a weighted connected undirected graph without negative cycles, $T_r = (V, E', r, \nu)$ the SPT of G rooted in $r \in V$, $G' = (V, E^{\tau'}_{[1..4]}, r, \mu')$ the SPGT and $G'' = (V, E^{\tau''}_{[1..4]}, r, \mu)$ the GSPT both derived from G by means of T_r and starting from r. The *Generalised shortest PathS Tree* (GPST) $G''' = (V, E^{\tau}_{[1..5]}, r, \mu)$ is derived from G' and G'' by refining the edge typing functions τ' and τ'' in terms of $\tau\colon E \to \{\mathrm{c}, \mathrm{k}, \mathrm{r}, \mathrm{s}, \mathrm{v}\} = \mathcal{T}$ such that

$$\forall e \in E\colon \begin{cases} \tau''(e) \in \{\mathrm{k}, \mathrm{r}, \mathrm{v}\} & \Rightarrow \quad \tau(e) = \tau''(e) & \\ \tau(e) = \mathrm{c} & \Rightarrow \quad e \in E^{\tau''}_{[4]} \setminus E^{\tau'}_{[4]} & \textit{(cross-reference edges)} \\ \tau(e) = \mathrm{s} & \Rightarrow \quad e \in E^{\tau'}_{[4]} & \textit{(short cut edges)} \end{cases}$$

Further, we set $E^{\tau}_{[1]} = \{e \in E \mid \tau(e) = \mathrm{k}\} = E^{\tau'}_{[1]} = E^{\tau''}_{[1]} = E', E^{\tau}_{[2]} = \{e \in E \mid \tau(e) = \mathrm{v}\} = E^{\tau''}_{[2]}, E^{\tau}_{[3]} = \{e \in E \mid \tau(e) = \mathrm{r}\} = E^{\tau''}_{[3]}, E^{\tau}_{[4]} = \{e \in E \mid \tau(e) = \mathrm{c}\}, E^{\tau}_{[5]} = \{e \in E \mid \tau(e) = \mathrm{s}\} = E^{\tau'}_{[4]}$ where $E^{\tau}_{[4]} \cup E^{\tau}_{[5]} = E^{\tau''}_{[4]}$. ◇

Corollary 1.8 *Apart from short cut edges $e \in E^{\tau}_{[5]}$, shortest paths in a GPST G solely contain kernel edges.*

This corollary is a simple consequence of Theorem 1.3 and the fact that GPSTs basically induce a partition of lateral edges into the subset of short cut edges $e \in E^{\tau}_{[5]}$ which contribute to shortest paths and the subset of cross-reference edges $e \in E^{\tau}_{[4]}$ which do not. Based on Definition 1.10 and the latter corollary we can now establish a fine-grained semantics of lateral edges in addition to that of vertical edges induced by Theorem 1.3:

- *Short cut edges:* In order to span shortest paths among vertices subject to the topology of the underlying graph G, an SPGT G' selects a subset of lateral edges. Lateral edges in the GSPT G'' of G with the same kernel as G' which do not shorten paths in the latter sense are excluded from G'. Thus, in SPGTs lateral edges are genuine short cuts: they are the only means to establish shortest paths apart from the corresponding kernel tree. Think of such edges, for example, as short cuts in small world-graphs [66] which are used to establish a high cluster value within the network. In contrast to this, vertical edges realise more costly aggregations or, in terms of semiotics, conceptual condensations along chains of hypernymy relations.
- *Cross-reference edges:* Compared with short cut edges and vertical edges, cross-reference edges neither aggregate nor shorten any (e.g. conceptual) relation in any other way. They rather serve as *transverse* edges which bridge weakly related regions of a generalised tree. That is, cross-reference edges build short cuts in a broader sense as they bridge more distant vertices of the underlying graph G. In terms of small world-graphs, cross-reference edges correspond to somehow randomly rewired (and therefore costly) edges. They provide short average geodesic distances and, thus, connectivity even among less related vertices as a precondition of efficient information flow within the network [66] — however, at the cost of a loss in coherence among the vertices linked in such a manner.

Now we are in a position to directly interpret the semantics of kernel and peripheral edges in terms of real semiotic systems. Assume, for example, that we model a social (terminological [54]) ontology [55] as spanned by the category system of the Wikipedia. In this case, vertical edges can be used to model transitivity among hypernymy relations, while lateral links are a means to map co-classifications or polymorphic categorisations. Obviously, the role of vertical and lateral edges is quite different in this example so that this distinction should be reflected in the built-up of a GT spanning the social ontology. From a *functional* point of view, this distinction can be emphasised as follows:

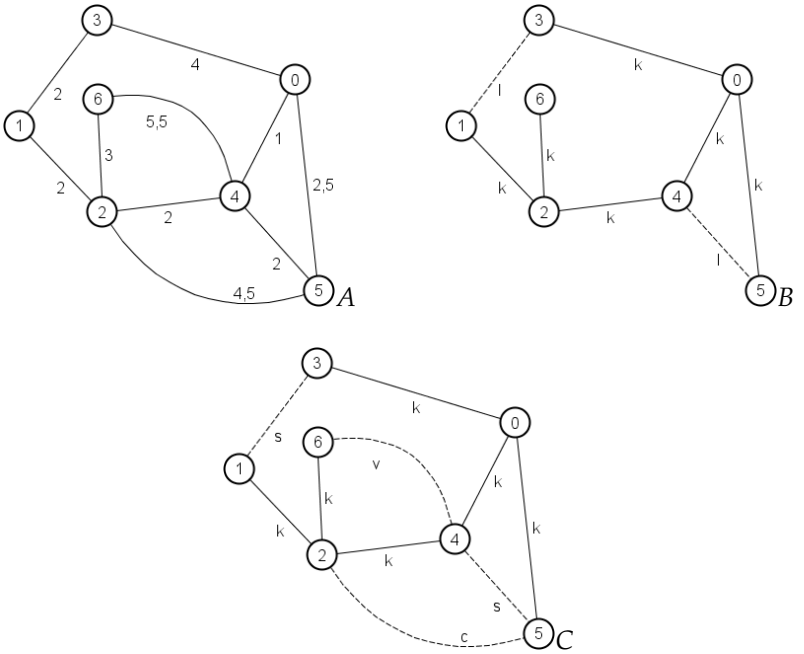

Fig. 1.5: A graph A together with its SPGT (graph B) rooted in vertex 0 and the corresponding GPST (graph C). For reasons of simplification, edge weights are omitted in the graphical representations of graph B and C.

- *Searching:* In order to efficiently search or walk through a network, kernel edges are the first choice.
- *Changing:* If in contrast to the latter search function agents aim at non-randomly changing the current standpoint (view or topic) in the course of their network transitions, they should select lateral or, more specifically, short cut edges as far as they are available. A more random walk through the network is instead of this supported by following cross-reference links.
- *Abridging:* Finally, in order to cut short the traversal of the kernel hierarchy, vertical edges are the primary means.

This functional scenario gives a complete and distinguished semantics of the different types of edges in generalised trees and, therefore, motivates the introduction of this kind of tree-like graphs in-between complete order — as manifested by trees — and randomness — as manifested by random graphs. That is, unlike the literature about generalised trees and complex networks introduced so far we are now in a position in which we can assign edges a certain role as a function of their contribution to the topology of the network in which they are spanned. According to the definition of GPSTs, this sys-

tem of potential roles of edges distinguishes five types. Obviously, this goes much beyond present-day approaches to complex networks in which edges are normally neither labeled nor typed.

Let us now consider an example which exemplifies the notion of a GPST:

Example *Let the graph $A = (V, E, \mu)$ in Figure 1.5 be given. Then, graph $B = (V, E^{\tau_B}_{[1..4]}, 0, \mu_B)$ is an SPGT spanned over A by means of the kernel SPT $T_0 = (V, \{\{0,3\},\{0,4\},\{0,5\},\{2,4\},\{1,2\},\{2,6\}\}, 0)$ and the set of shortest paths-inducing lateral edges $E^{\tau_B}_{[4]} = \{\{1,3\},\{4,5\}\}$. Next, graph $C = (V, E^{\tau_C}_{[1..5]}, 0, \mu_C)$ is a GPST with the following sequence of edge sets: $E^{\tau_C}_{[1]} = E^{\tau_B}_{[1]}$, $E^{\tau_C}_{[2]} = \{\{4,6\}\}$, $E^{\tau_C}_{[3]} = \emptyset$, $E^{\tau_C}_{[4]} = \{\{2,5\}\}$, $E^{\tau_C}_{[5]} = E^{\tau_B}_{[4]}$.*

So far, we have introduced generalised trees by combining the context-sensitive formation of kernel trees with a more and more constrained semantics of peripheral edges. This approach goes beyond existing efforts to use generalised trees as a graph model in-between tree-like structures and unconstrained graphs. The reason is that it does not only demarcate generalised from ordinary graphs by means of typed edges. These types are also justified in functional terms which are missing in general graphs. Section 1.2.8 shows that this extension is a prerequisite of a graph model of a certain class of semiotic systems. But before introducing this model we extend our approach by *orientating* generalised trees.

1.2.7 Accounting for Orientation: Directed Generalised Trees

So far we have considered only undirected graphs. As is well-known from graph theory things look quite different when dealing with oriented graphs. The MST of a directed graph, for example, cannot be computed in the same way as its undirected counterpart [58]. In the remainder of this section we provide an orientation of generalised trees. However, we concentrate on complexity statements leaving the proofs of many counterparts of the theorems presented above for future work.

Definition 1.11 (Directed Generalised Tree) Let $T = (V, A', r)$ be a directed tree rooted in $r \in V$. Let further $P_{rv} = (v_{i_0}, a_{j_1}, v_{i_1}, \ldots, v_{i_{n-1}}, a_{j_n}, v_{i_n}), v_{i_0} = r, v_{i_n} = v, a_{j_k} \in A', in(a_{j_k}) = v_{i_{k-1}}, out(a_{j_k}) = v_{i_k}$, $1 \leq k \leq n$, be the unique path in T from r to v and $V(P_{rv}) = \{v_{i_0}, \ldots, v_{i_n}\}$ the set of all vertices of P_{rv}. A *Directed Generalised Tree* (DGT)

$$G = (V, A, \tau, r)$$

induced by T is a pseudograph (i.e. a multigraph possibly with multiple and parallel arcs or loops) whose arcs are typed by the function $\tau : A \to \{\mathrm{d}, \mathrm{k}, \mathrm{l}, \mathrm{r}, \mathrm{u}\}$ as follows:

$$\forall a \in A\colon \begin{cases} \tau(a) = \mathrm{k} & \Rightarrow & a \in A_{\mathrm{k}} = A' \\ & & \textit{(kernel arcs)} \\ \tau(a) = \mathrm{u} & \Rightarrow & a \in A_{\mathrm{u}} = \{a \,|\, in(a) = v \in V \\ & & \wedge\, out(a) = w \in V(P_{rv}) \setminus \{v\}\} \\ & & \textit{(upward arcs)} \\ \tau(a) = \mathrm{d} & \Rightarrow & a \in A_{\mathrm{d}} = \{a \,|\, in(a) = w \in V(P_{rv}) \setminus \{v\} \\ & & \wedge\, out(a) = v \in V\} \\ & & \textit{(downward arcs)} \\ \tau(a) = \mathrm{r} & \Rightarrow & a \in A_{\mathrm{r}} = \{a \,|\, in(a) = out(a) = v \in V\} \\ & & \textit{(reflexive arcs)} \\ \tau(a) = \mathrm{l} & \Rightarrow & a \in V^2 \setminus (A_{\mathrm{k}} \cup A_{\mathrm{u}} \cup A_{\mathrm{d}} \cup A_{\mathrm{r}}) \\ & & \textit{(lateral arcs)} \end{cases}$$

such that $A^\tau_{[1]} = \{a \in A \,|\, \tau(a) = \mathrm{k}\}, A^\tau_{[2]} = \{a \in A \,|\, \tau(a) = \mathrm{u}\}, A^\tau_{[3]} = \{a \in A \,|\, \tau(a) = \mathrm{d}\}, A^\tau_{[4]} = \{a \in A \,|\, \tau(a) = \mathrm{r}\}, A^\tau_{[5]} = \{a \in A \,|\, \tau(a) = \mathrm{l}\}$ is a partition of A such that $A = \cup_{i=1}^5 A^\tau_{[i]}$ and $\forall 1 \leq i < j \leq 5 : A^\tau_{[i]} \cap A^\tau_{[j]} = \emptyset$. Because of the interdependence of τ and the latter partition we alternatively denote G by $(V, A^\tau_{[1..5]}, r)$ where $a \in A^\tau_{[1..5]}$ iff $a \in \cup_{i=1}^5 A^\tau_{[i]}$. In other words, directed generalised trees G are interchangeably noted as (V, A, τ, r) and $(V, A^\tau_{[1..5]}, r)$. We say that G is *generalised by its lateral, reflexive, upward and downward arcs*. The directed generalised tree $G = (V, A, \tau, r)$ induces the directed tree $\mathrm{kern}(G) = (V, A^\tau_{[1]}, r) = T$ called *kernel (tree)* or *skeleton* of G. Further, the graph $\mathrm{periphery}(G) = (V, \cup_{i=2}^5 A^\tau_{[i]})$ is called *periphery* or *complementary graph* of G. Arcs belonging to $\mathrm{periphery}(G)$ are called *peripheral arcs* (complementing the set of kernel arcs). Finally, any generalised tree (V, A, τ, r, μ) with the arc weighting function $\mu\colon A \to \mathbb{R}$ is called a *weighted undirected generalised tree*. ◇

Remark. A simple consequence of orientating GTs is the need to distinguish between upward and downward arcs which both are subsumed under the notion of vertical edges in undirected GTs.

Remark. By analogy to undirected GTs we henceforth assume that μ is a function from A to $\mathbb{R}^+ \setminus \{0\}$. Once more, the reason is to prevent negative cycles. Further, we only deal with finite graphs.

Definition 1.12 (Directed Generalised Spanning Tree) Let $G = (V, A, \mu)$ be a weighted connected digraph without negative cycles. Let further $r \in V$ be a vertex in G. A *Directed Generalised Spanning Tree* (DiGST) $G' = (V, A^\tau_{[1..5]}, r, \mu)$ of G is a directed generalised tree whose kernel $\mathrm{kern}(G') = T$ is a directed spanning tree $T = (V, A^\tau_{[1]}, r, \nu)$ of G rooted in r such that $A = \cup_{i=1}^5 A^\tau_{[i]}$ and

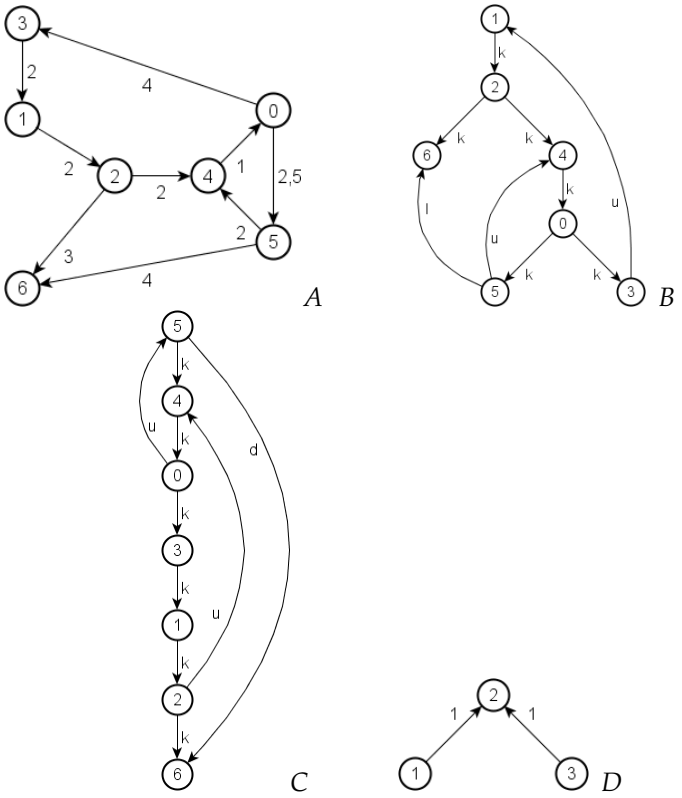

Fig. 1.6: A digraph A and two directed spanning trees derived from it: rooted in vertex 1 (graph B) and alternatively rooted in vertex 5 (graph C). For reasons of simplification, edge weights are omitted in graph B and C.

ν is the restriction of μ to $A^{\tau}_{[1]}$. We say that G' *is spanned over* G *by means of* T *starting from* r. G is called the *underlying graph* of G'. ◇

Example *Let the digraph A in Figure 1.6 be given. In this case, we have at most six different directed spanning trees (vertex 6 does not root a spanning tree). Starting from the graph D in Figure 1.6 we see that it does not induce any DGT since D does not have any directed spanning tree.*

Theorem 1.4 *Given a weighted connected digraph $G = (V, A, \mu)$ without negative cycles, a vertex $r \in V$ and a spanning tree $T = (V, A', r, \nu)$ of G rooted in r. Then, the time complexity of computing the directed generalised spanning tree $G' = (V, A^{\tau}_{[1..5]}, r, \mu)$ spanned over G by means of T is in the order of $\mathcal{O}(|V| + |A|)$.*

Proof. By analogy with the proof of Theorem 1.1 we observe that solving this task demands differentiating between upward, downward and lateral arcs. The reason is that while kernel arcs are identified by their membership

Algorithm 2 Spanning Peripheral Arcs

Require: A digraph $G = (V, A, \mu)$, a spanning tree $T = (V, A', r, \nu)$ of G and a vertex $r \in V$ according to Definition 1.12.

Ensure: The set $A^\tau_{[2]}$ of upward, the set $A^\tau_{[3]}$ of downward, the set $A^\tau_{[4]}$ of reflexive and the set $A^\tau_{[5]}$ of lateral arcs of the DiGST G' spanned over G by means of T starting from r.

```
1:  procedure SpanningPeripheralArcs(G, T, r)
2:      A^τ_[1] ← A'; A^τ_[2] ← A^τ_[3] ← A^τ_[4] ← A^τ_[5] ← ∅
3:      x ← VectorOfAllPathsInTreeStartingFromRoot(T, r)
4:      for a ∈ A \ A^τ_[1] do
5:          v ← in(a), w ← out(a)
6:          if v = w then
7:              A^τ_[4] ← A^τ_[4] ∪ {a}
8:          else
9:              v ← x[v] ∧ w ← x[w]
10:             if v[w] then
11:                 A^τ_[2] ← A^τ_[2] ∪ {a}
12:             else if w[v] then
13:                 A^τ_[3] ← A^τ_[3] ∪ {a}
14:             else
15:                 A^τ_[5] ← A^τ_[5] ∪ {a}
16:             end if
17:         end if
18:     end for
19:     return A^τ_[2..5]
20: end procedure
```

to T, reflexive arcs are distinguished by the fact that they contain the same vertex twice. Because of the definition of lateral arcs this further means that we have to decide whether a given arc $a \in (A^\tau_{[1..5]} \setminus A^\tau_{[1]}) \setminus A^\tau_{[4]}$ is an upward or a downward arc. This decision is computed by Algorithm 2 by analogy to Algorithm 1. The only difference is that we distinguish between upward and downward arcs (using the same format and method). Thus, the time effort of Algorithm 2 is basically induced by the Lines 5–17 which are repeated exactly $|A| - |A'|$ times so that because of the constant complexity of the latter operations the order in question is $\mathcal{O}(|A| - |A'|) = \mathcal{O}(|A|)$. Further, the complexity of performing a breadth-first search (Line 3) is, as before, of order $\mathcal{O}(|V| + |E|) = \mathcal{O}(|V| + |V| - 1) = \mathcal{O}(|V|)$ so that we get $\mathcal{O}(|V| + |A|)$ as the desired upper bound. Again, more efficient algorithms can be envisioned

but are out of the focus of this chapter as Algorithm 2 is already sufficiently efficient. □

Now we can introduce and exemplify directed MSGTs as follows:

Definition 1.13 (Directed Minimum Spanning Generalised Tree) Let $G = (V, E, \mu)$ be a weighted connected digraph, $T = (V, A', r, \nu)$ a minimum spanning tree of G rooted in some $r \in V$. The *Directed Minimum Spanning Generalised Tree* (DiMSGT) induced by T is a directed generalised tree $G_{T_r} = (V, A^{\tau}_{[1..5]}, r, \mu)$ spanned over G by means of the kernel tree T starting from r. ◇

Example *Let the digraph $A = (V, E, \mu)$ in Figure 1.7 be given. Then, digraph $B = (V, A^{\tau_B}_{[1..5]}, 2, \mu_B)$ is a DiMSGT spanned over A by means of the kernel MST $T_2 = (V, A', 0)$, $A' = \{(2,4), (2,6), (4,0), (0,5), (0,3), (3,10), (3,1), (10,12), (1,7), (1,8), (12,11), (8,9)\}$, together with the following partition of the arc set $A^{\tau_B}_{[1..5]}$: $A^{\tau_B}_{[1]} = A'$, $A^{\tau_B}_{[2]} = \{(1,2), (5,4)\}$, $A^{\tau_B}_{[3]} = \{(3,11), (10,11), (3,7)\}$, $A^{\tau_B}_{[4]} = \emptyset$, $A^{\tau_B}_{[5]} = \{(7,9)\}$. Note that the subgraph spanned by the vertices 1, 7, 8 and 9 is a typical case which demarcates Prim's algorithm of spanning MSTs from the corresponding algorithms adapted to digraphs. While Prim's algorithm would select the arc $(1,7)$, then the arc $(7,9)$ and finally the arc $(1,8)$, we realise that the choice of $(1,7)$, $(1,8)$ and $(8,9)$ is less costly [68].*

Corollary 1.9 *Because of Theorem 1.4 the time complexity of generating a DiMSGT is in the order of $\mathcal{O}(|V| + |E| + \min\{|E| \log |V|, |V|^2\})$ when using a standard algorithm [58] to generate a directed minimum spanning tree as its kernel.*

This corollary is a simple consequence of separating the generation of spanning trees from spanning peripheral arcs as realised by Algorithm 2. As a DiMSGT rooted in a preselected vertex r equals the shortest path tree rooted in the same vertex it is superfluous to consider *directed* generalised shortest path trees. As an alternative we consider *Directed Generalised Dependency Trees* (DiGDT). These are generalised trees which are spanned by means of directed dependency trees which, in turn, result from orientating so called dependency trees. *Dependency Tree*s (DT) have been used in computational linguistics to order association data in a tree-like fashion [35, 38, 49]. They are generated as follows: for a distinguished vertex r of a graph $G = (V, A, \mu)$, vertices are inserted into the DT rooted by r in ascending order of their geodesic distance from r where the predecessor of any vertex v to be inserted is chosen to be the vertex w which in terms of μ is most close to v among all vertices already inserted into the DT. Look at Figure 1.7 and vertex 2 as our distinguished root vertex. In this case, we realise first that the vertices of graph A are inserted into a DT rooted by 2 according to their geodesic distance to 2. Thus, we get the following sequence of vertices to be inserted: $4, 6, 0, 5, 3, 1, 10, 11, 12, 7, 8, 9$. Based on this sequence we get a kernel spanning DT as shown in Figure 1.7 by graph

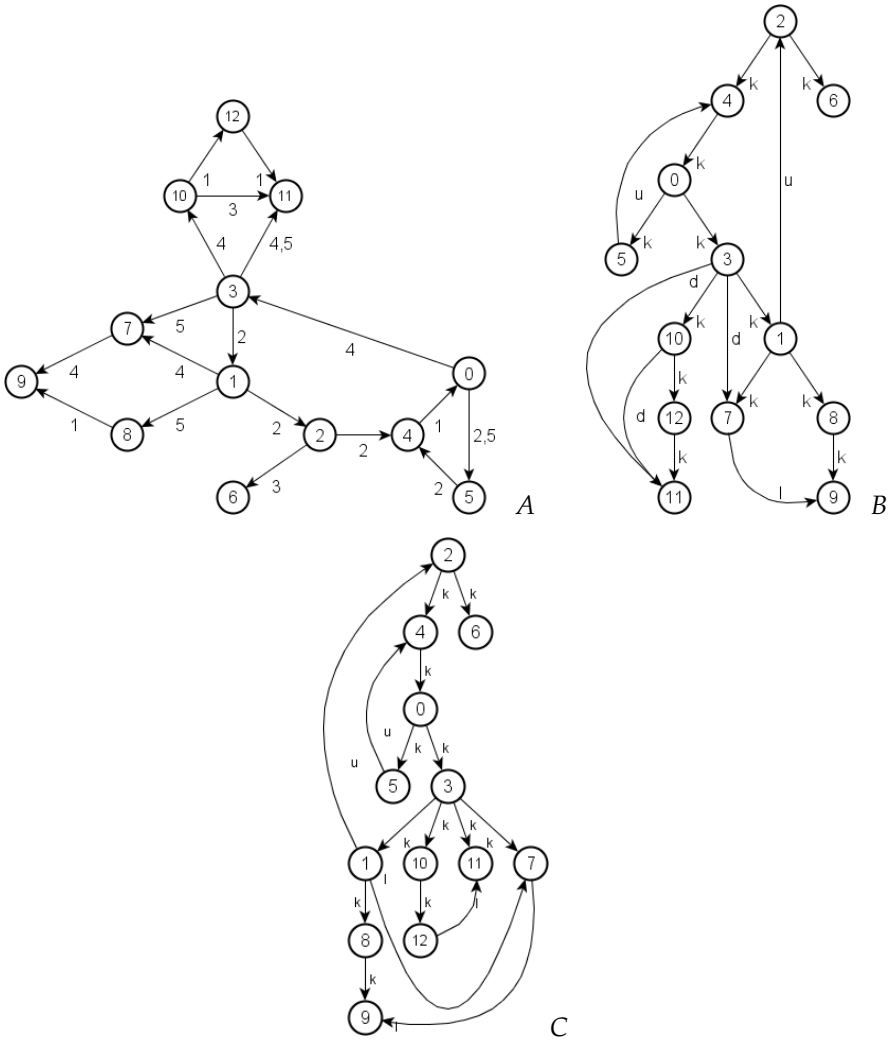

Fig. 1.7: A directed graph A together with its DiMSGT (graph B) rooted in vertex 2 and a directed generalised dependency tree (graph C) rooted in the same vertex. For reasons of simplification, arc weights are omitted in the graphical representations of graph B and C.

C which is a *generalised* dependency tree spanned by means of the latter DT starting from 2. The generation of this DiGDT is more context-sensitive than the one of the corresponding DiMSGT since the choice of the root uniquely determines the sequence in which vertices are processed, that is, according to their order of being *primed* by the root vertex. As a consequence, DiGDTs realise a sort of construction-integration process [33] in which the root vertex initiates firstly a process of spreading activation or information percolation

which is secondly organised in the form of a generalised tree. In other words: in Figure 1.7, the DiGDT C structures the network given by graph A more from the perspective of vertex 2 than this is done by the DiMSGT B. We do not approach in formalising this notion here but hint at a publication in which dependency trees and more general Markov trees are studied as the kernel trees of generalised trees in detail — cf. [41].

At this stage, we may envision a lot of theorems about directed generalised trees by analogy to those proved for their undirected counterparts. However, we resist following this branch of research and go back to elaborating the framework of undirected generalised trees — this time with a strict view of semiotic modeling.

1.2.8 Generalised Trees, Quality Dimensions and Conceptual Domains

So far we have introduced generalised trees as a fairly expressive, nonetheless well-constrained model in-between the extremal cases of trees and general graphs. In this section, we explore the representational potential of generalised trees a step further. As done in the sections before, we do that in graph-theoretical terms. The general story behind this approach is that we seek a model beyond *semantic spaces* as far as they are based on a purely geometric understanding of meaning relations [10,30,32,49,50]. Although we agree with the conception of usage-based semantics and its quantitative reconstruction by semantic spaces, we quarrel with the space complexity of this model and — as a result of this — with its cognitive implausibility. Without going into the details of this argumentation we just mention that semantic spaces equal completely connected graphs as their meaning points are always directly relatable in terms of their distance without the need of considering intermediate points. Consequently, they always have a maximal cluster value [66] — far away from what is known about real semiotic networks [39,56] and their small world-like topology. As an alternative to this undesirable state we seek a less compact model with a realistically sparse topology in conjunction with a tree-like skeleton by analogy to conceptual hierarchies.

In order to approach this model we refer to [27] who elaborates *conceptual spaces* as a level of representation in-between sub-symbolic association networks of lower resolution and symbolic models of higher resolution. Roughly speaking, a conceptual space is based on a system of conceptual domains which are *integral dimensions* used to map points onto the space. That is, for a set of quality dimensions $\{D_1, \ldots, D_n\}$ one can build a conceptual space in which objects v to be observed are represented as vectors $\mathbf{v}' = (v(D_1), \ldots, v(D_n))$ where $v(D_i), 1 \leq i \leq n$, is the value taken by object v on dimension D_i. A central starting point of [27] is to view domains as

systems of interrelated quality dimensions. This gives a conceptual space an internal structure as its objects can be characterised by *structured values* which they take on the corresponding domain.[4]

[27] does not completely determine the mathematical notion of a conceptual space but relies on an axiomatic approach by naming necessary conditions of candidate implementations. In this sense, semantic spaces are just one way of implementing conceptual spaces which leave plenty room for developing alternative, topologically more constrained space models. This is exactly our gateway to make a first step in promoting generalised trees as such an alternative. In this section we show how generalised trees can be conceived as conceptual domains. That way, we open the door to less complex representation formats apart from semantic spaces, formats which provide the efficiency of tree-like structures together with the structural freedom of networks. In order to approach this goal we proceed as follows: Firstly, we define a metric space based on generalised trees. Secondly, we interrelate this definition with the notion of betweenness and equidistance in terms of generalised trees. Thirdly, we introduce an interpretation of generalised trees as a sort of conceptual domain by which conceptual spaces are spanned as networks of networks (cf. Section 1.2.9). Once more, it turns out that generalised shortest path trees are the valuable starting point of this endeavour:

Corollary 1.10 *Let $G = (V, E, \mathcal{L}_V, \mathcal{L}_E, \mu)$ be a weighted connected graph according to Definition 1.1, $G' = (V, E^{\tau'}_{[1..4]}, r, \mu)$ be a GSPT and $G'' = (V, E^{\tau''}_{[1..5]}, r, \mu)$ a GPST spanned over G by means of the shortest path tree T_r starting from $r \in V$. Then, $\hat{\mu}'$ (see Definition 1.2) is a distance function in $\hat{G}' = (V, E^{\tau'}_{[1..3]}, r, \mu')$ and $\hat{\mu}''$ a distance function in $\hat{G}'' = (V, E^{\tau''}_{[1..4]}, r, \mu'')$ — μ' and μ'' are the restrictions of μ to $E^{\tau'}_{[1..3]}$ and $E^{\tau''}_{[1..4]}$, respectively. That is, $(\hat{G}', \hat{\mu}')$ and $(\hat{G}'', \hat{\mu}'')$ are metric spaces.*

Proof. Because of Theorem 1.3 and Corollary 1.8 we can concentrate on kernel edges when considering shortest paths and geodesic distances. As a trivial consequence of these two theorems, Corollary 1.10 is reduced to a statement about trees since neither vertical nor cross-reference links interfere with the function of kernel links, that is, establishing shortest paths. Thus, the proof simply looks as follows:

- *Minimality:* The geodesic distance between two vertices is a non-negative, real-valued function which because of the definition of $\hat{\mu}$ is 0 in the case of two vertices v, w if and only if $v = w$ (see Definition 1.2).
- *Symmetry:* The symmetry of $\hat{\mu}'$ and $\hat{\mu}''$ simply follows from the fact that we deal with undirected graphs.
- *Triangle inequality:* If P is the geodesic path between u and v (which because of Definition 1.2 is uniquely defined) and P' the geodesic path be-

4) For more details of this notion see [27].

tween v and w, then $P \circ P'$ is the geodesic path between u and w so that $\hat{\mu}'(u,v) + \hat{\mu}'(v,w) = \hat{\mu}'(u,w)$ — as claimed by Theorem 1.3 this does not interfere with any vertical edge. In the case of $\hat{\mu}''$ we have to argue analogously.

□

Following [27], we now define the relation of betweenness and the relation of equidistance in terms of generalised trees where the latter form — according to Corollary 1.10 — a special kind of metric space:

Definition 1.14 (Geodesic Betweenness) Let $G = (V, E, \mathcal{L}_V, \mathcal{L}_E, \mu)$ be a weighted connected graph according to Definition 1.1. Then, we define the relation $B \subseteq V^3$ where

$$\forall u, v, w \in V : B(u,v,w) \Leftrightarrow u \neq v \neq w \wedge v \in V(GP_\mu(u,w))$$

The relation B is called *Relation of Geodesic Betweenness*. This is the relation of all vertices u, v, w for which v is on the geodesic path between u and w. Every vertex v for which $B(u,v,w)$ is called *geodesically between* u and w. ◇

Corollary 1.11 *Let $G = (V, E, \mathcal{L}_V, \mathcal{L}_E, \mu)$ be a weighted connected graph according to Definition 1.1, $G' = (V, E^{\tau'}_{[1..4]}, r, \mu)$ be a GSPT and $G'' = (V, E^{\tau''}_{[1..5]}, r, \mu)$ a GPST spanned over G by means of the shortest path tree T_r starting from $r \in V$. Then, firstly, the relation B satisfies the* Axioms *B1–B4* of Betweenness *[27] in $\hat{G}' = (V, E^{\tau'}_{[1..3]}, r, \mu')$ and in $\hat{G}'' = (V, E^{\tau''}_{[1..4]}, r, \mu'')$. Secondly, for any $u, v, w \in V : (B(u,v,w) \Leftrightarrow \hat{\mu}'(u,v) + \hat{\mu}'(v,w) = \hat{\mu}'(u,w)) \wedge (B(u,v,w) \Leftrightarrow \hat{\mu}''(u,v) + \hat{\mu}''(v,w) = \hat{\mu}''(u,w))$. μ' and μ'' are the restrictions of μ to $E^{\tau'}_{[1..3]}$ and $E^{\tau''}_{[1..4]}$, respectively.*

Proof. Trees are known to satisfy the axioms of betweenness. By Theorem 1.3 and Corollary 1.8 we know that neither vertical edges in GSPTs nor vertical and cross-reference edges in GPSTs interfere with kernel edges in spanning geodesic paths. Thus, the first part of Corollary 1.11 reduces to a well-known statement about (kernel) trees so that we can claim that B satisfies the following axioms of betweenness:

- *B*1: $\forall u, v, w \in V : u \neq v \neq w \Rightarrow (B(u,v,w) \Rightarrow B(w,v,u))$.
- *B*2: $\forall u, v, w \in V : u \neq v \neq w \Rightarrow (B(u,v,w) \Rightarrow \neg B(v,u,w))$.
- *B*3: $\forall v, w, x, y \in V : v \neq w \neq x \neq y \Rightarrow (B(v,w,x) \wedge B(w,x,y) \Rightarrow B(v,w,y))$.
- *B*4: $\forall v, w, x, y \in V : v \neq w \neq x \neq y \Rightarrow (B(v,w,y) \wedge B(w,x,y) \Rightarrow B(v,w,x))$.

The second part of Corollary 1.11 simply interrelates the triangle inequality of Corollary 1.10 with the present corollary. □

By analogy with Definition 1.14 and Corollary 1.11 we get the following definition of equidistance in conjunction with its concomitant corollary:

Definition 1.15 (Geodesic Equidistance) Let $G = (V, E, \mathcal{L}_V, \mathcal{L}_E, \mu)$ be a weighted connected graph according to Definition 1.1. Then, we define the relation $E \subseteq V^4$ where

$$\forall v, w, x, y \in V : E(v, w, x, y) \quad \Leftrightarrow \quad v \neq w \neq x \neq y \wedge \mu(GP_\mu(v, w)) = \mu(GP_\mu(x, y))$$

The relation E is called *Relation of Geodesic Equidistance*. This is the relation of all vertices v, w, x, y for which the geodesic distance between v and w equals the geodesic distance between x and y. Any two pairs of vertices v, w and x, y for which $E(v, w, x, y)$ are called *geodesically equidistant*. ◇

This allows us to formulate the following self-evident corollary:

Corollary 1.12 *Let $G = (V, E, \mathcal{L}_V, \mathcal{L}_E, \mu)$ be a weighted connected graph according to Definition 1.1, $G' = (V, E^{\tau'}_{[1..4]}, r, \mu)$ be a GSPT and $G'' = (V, E^{\tau''}_{[1..5]}, r, \mu)$ a GPST spanned over G by the shortest path tree T_r starting from $r \in V$. Then, firstly, the relation E satisfies the* Axioms *E*1–*E*4 of Equidistance *[27] in $\hat{G}' = (V, E^{\tau'}_{[1..3]}, r, \mu')$ and in $\hat{G}'' = (V, E^{\tau''}_{[1..4]}, r, \mu'')$. Secondly, for any $v, w, x, y \in V : (E(v, w, x, y) \Leftrightarrow \hat{\mu}'(v, w) = \hat{\mu}'(x, y)) \wedge (E(v, w, x, y) \Leftrightarrow \hat{\mu}''(v, w) = \hat{\mu}''(x, y))$. μ' and μ'' are the restrictions of μ to $E^{\tau'}_{[1..3]}$ and $E^{\tau''}_{[1..4]}$, respectively.*

Proof. Once more, the only thing we need to hint at is that the generalised sub-trees $\hat{G}'$ and $\hat{G}''$ do not contain any edges which interfere with their kernel edges in spanning geodesic paths. Thus, we can claim that E meets the following axioms of equidistance within these generalised subtrees:

- *E*1: $\forall u, v, w \in V : E(u, u, v, w) \Rightarrow v = w$.
- *E*2: $\forall v, w \in V : E(v, w, w, v)$.
- *E*3: $\forall u, v, w, x, y, z \in V : u \neq v \neq w \neq x \neq y \neq z \Rightarrow (E(u, v, w, x) \wedge E(u, v, y, z) \Rightarrow E(w, x, y, z))$.
- *E*4: $\forall u, v, w, x, y, z \in V : u \neq v \neq w \neq x \neq y \neq z \Rightarrow (B(u, v, w) \wedge B(x, y, z) \wedge E(u, v, x, y) \wedge E(v, w, y, z) \Rightarrow E(u, w, x, z))$.

□

By Corollary 1.11 and 1.12 we see that well-defined generalised sub-trees of GSPTs and GPSTs respect the axioms of betweenness and equidistance. Moreover, by Corollary 1.10 we additionally see that we get a simple metric on instances of these classes of graphs. According to [27] these are basic structural constraints to be satisfied by the dimensions of conceptual spaces. In other words, it seems plausible to build conceptual spaces in terms of a special kind of generalised tree — this relates especially to GPSTs. By their hi-

erarchical skeleton, these trees respect basic constraints of conceptual spaces but nevertheless share the full expressiveness of graphs. This paves the way for a graph-theoretical model of conceptual structures beyond ordinary trees and below the space complexity of semantic spaces. *But what does it mean to use generalised trees for spanning conceptual spaces?* More specifically: *How can we think of generalised trees as quality dimensions?* In order to answer these questions we utilise the notion of a conceptual domain as a system of interrelated or interdependent quality dimensions. More specifically, we define a conceptual space as a set of quality dimensions with a topological structure as defined by generalised trees where a single domain equals a generalised tree as a kind of structured dimension. In other words, the vertices of a GSPT or a GPST, respectively, define basic dimensions which by virtue of their edges form integral dimensions. The integrity of the dimensions is reflected by the hierarchical skeleton of the respective GT. As a consequence, objects o to be mapped onto a conceptual domain $\mathcal{D}$ are interrelated with a subset of vertices of $\mathcal{D}$ such that all edges generated by this measurement operation preserve the generalised tree-like topology of $\mathcal{D}$. That way, a measurement of o along a basic dimension v of $\mathcal{D}$ equals the geodesic path ending at v and o as a result of adding o as a new vertex to $\mathcal{D}$ subject to preserving the structural constraints of this generalised tree. In Section 1.3 we exemplify this structuralistic notion of measurement in detail. Note that so far we have sketched GT-based conceptual spaces only in terms of undirected graphs leaving the examination of conceptual spaces based on *directed* GTs for future work. The next section shows how this notion of a graph-like representation of conceptual structures is extended in order to cope (in Section 1.3) with distributed knowledge as provided by social ontologies and social encyclopedias.

1.2.9 Generalised Forests as Multi-Domain Conceptual Spaces

So far we have considered generalised trees as intermediary units between trees and graphs. We have also related this notion to cognitive modeling in terms of conceptual domains as introduced by [27]. In this section we approach this perspective of spanning conceptual spaces by generalised trees a step further. This is done by interlinking conceptual domains as separable, inherently structured dimensions whose values — taken by objects to be mapped onto a given conceptual space — are measured separately from each other. From a graph-theoretical point of view, interlinked domains can be modelled by appropriately generalising the notion of a forest (of trees). Following this approach, we build conceptual spaces in the form of generalised forests of generalised trees. This is done as follows:

Definition 1.16 (Generalised Forest) A *Generalised Forest* (GF) is a graph $G = (V, E^{\tau}_{[0..4]}, \mu)$ such that the connected components $D_1, \ldots, D_n$ of the subgraph $G' = (V, E^{\tau}_{[1..4]}, \mu')$ of G are generalised trees $D_i = (V_i, E^{\tau_i}_{[1..4]}, r_i, \mu_i)$, $1 \leq i \leq n$, which satisfy the following structural constraints:

1. The sequence $V_1, \ldots, V_n$ is a partition of V, i.e., $V = \cup_{i=1}^{n} V_i$ and $\forall 1 \leq i < j \leq n \colon V_i \cap V_j = \emptyset$. In order to denote this partition we use the function $\mathrm{V} \colon V \to \{V_1, \ldots, V_n\}$ where $\forall v \in V, \forall 1 \leq i \leq n \colon \mathrm{V}(v) = V_i \Leftrightarrow v \in V_i$.
2. The sequence $E^{\tau_1}_{[1..4]}, \ldots, E^{\tau_n}_{[1..4]}$ is a partition of $E^{\tau}_{[1..4]}$, that is, $E^{\tau}_{[1..4]} = \cup_{i=1}^{n} E^{\tau_i}_{[1..4]}$ and $\forall 1 \leq i < j \leq n : E^{\tau_i}_{[1..4]} \cap E^{\tau_j}_{[1..4]} = \emptyset$. Thus, $\forall 1 \leq i \leq n \colon \tau_i \subseteq \tau$.
3. $\tau : E^{\tau}_{[0..4]} \to \mathcal{T} = \{\mathrm{e}, \mathrm{k}, \mathrm{l}, \mathrm{r}, \mathrm{v}\}$ is an extended edge typing function such that $\forall e = \{v, w\} \in E^{\tau}_{[0..4]} \colon \tau(e) = \mathrm{e} \Rightarrow \mathrm{V}(v) \neq \mathrm{V}(w)$. That is, for any $1 \leq i \leq n \colon E^{\tau}_{[0]} \cap E^{\tau_i}_{[1..4]} = \emptyset$. Further, we define $E^{\tau}_{[0]} = \{e \in E^{\tau}_{[0..4]} \mid \tau(e) = \mathrm{e}\}$. Thus, $\forall \{v, w\} \in E^{\tau}_{[0]} \exists 1 \leq i < j \leq n \colon v \in V_i \wedge w \in V_j$. Edges $e \in E^{\tau}_{[0]}$ are called *external edges*.
4. μ_i, $1 \leq i \leq n$, denotes the restriction of μ to $E^{\tau_i}_{[1..4]}$.

In other words, a generalised forest is a graph which is partitioned into possibly interlinked generalised trees. We call the generalised trees D_i the *components* of the GF G linked by external edges as elements of $E^{\tau}_{[0]}$ and denote them by $\mathrm{cmp}(G) = \{D_1, \ldots, D_n\}$. ◇

Remark. A generalised forest is a generalised tree with an extended edge typing function which induces a decomposition of the underlying graph into a sequence of generalised trees by specifying external edges.

A connected graph with at least two vertices does not uniquely induce a generalised tree. Likewise, such a graph is not uniquely decomposable into the domains of a generalised forest. This simple observation gives plenty room for spanning generalised forests on a given graph subject to cost functions of external edges or the coherence of single domains. In this sense, we have to think of specific notions of generalised forests which are specialised by analogy to minimum spanning generalised trees, generalised shortest path trees and so on. We may think, for example, of the domains D of a GF G as subgraphs which span regions of higher "internal homogeneity" (e.g., in terms of graph clustering [61]) within the underlying graph while its external edges span a sort of minimum spanning tree over these domains. As a sample notion of this kind look at the following definition of conceptual graphs:

Definition 1.17 (Conceptual Graph) Let $G = (V, E, \mu)$ be a weighted connected graph and $\mu^* : V^2 \to \mathbb{R}_0^+$ a metric measuring the connectedness of vertices subject to the topology of G spanned by E. Other than μ, μ^* valuates directly as well as indirectly connected vertices. In this sense, it is reminiscent

of the notion of a transitive closure. However, we leave it open whether μ^* solely explores simple or even only shortest paths (as done by $\hat{\mu}$ — cf. Definition 1.2). Think of μ^*, e.g., as a function which measures the degree of unrelatedness or dissimilarity [9] of signs denoted by the vertices of G. Now a *Conceptual Graph* (CG) is a labeled generalised forest $G = (V, E^{\tau}_{[0..5]}, \zeta, r, \mathcal{L}_V, \mathcal{L}_E, , \mu)$ together with an edge typing function $\zeta \colon E^{\tau}_{[0]} \to \mathcal{T} = \{\mathrm{c}, \mathrm{k}, \mathrm{r}, \mathrm{s}, \mathrm{v}\}$ and the *top-level vertex* $r \in V$ which altogether satisfy the following additional constraints:

1. *Metric basic structure:* Each component $D_i \in \mathrm{cmp}(G)$ — henceforth called a *domain* of G — is a GPST — note that the index of $E^{\tau}_{[0..5]}$ is running from 0 to 5 so that the edge typing functions of the components of G distinguish among cross-reference and short cut edges.
2. *Micro level coherence:*

$$\forall e \in E^{\tau}_{[0]} \nexists e' \in E^{\tau}_{[1..5]} \colon \mu(e) < \mu(e')$$

3. *Meso level coherence:*

$$\forall v, w \in V \colon \mathrm{V}(v) \neq \mathrm{V}(w) \Rightarrow$$
$$\mu^*(v, w) > \max\{\max_{x,y \in \mathrm{V}(v)}\{\mu^*(x, y)\}, \max_{x,y \in \mathrm{V}(w)}\{\mu^*(x, y)\}\}$$

4. *Macro level structure:* $G' = (V', E', \zeta', R, \mu')$ is a GPST such that $V' = \{V_i | (V_i, E^{\tau_i}_{[1..5]}, r_i, \mu_i) \in \mathrm{cmp}(G)\}$, $|E'| = |E^{\tau}_{[0]}|$, $\forall e = \{v, w\} \in E^{\tau}_{[0]} : v \in V_i \wedge w \in V_j \wedge \chi_{E^{\tau}_{[0]}}(e) = k \Rightarrow \{V_i, V_j\} \in E' \wedge \chi_{E'}(\{V_i, V_j\}) = k \wedge \zeta'(\{V_i, V_j\}) = \zeta(e) \wedge \mu'(\{V_i, V_j\}) = \mu(e)$. Further, there exists an i such that $1 \leq i \leq |\mathrm{cmp}(G)|$ and $(V_i, E^{\tau_i}_{[1..5]}, r_i, \mu_i) \in \mathrm{cmp}(G)$ and $r \in R = V_i$. That is, r is the root of the root-building GPST-like component R of G.

We call G a $|\mathrm{cmp}(G)|$-*dimensional* conceptual graph. ◇

Remark. Satisfying Constraint 4 of Definition 1.17 guarantees the persistence of metric characteristics not only within single domains of a conceptual graph but also between the different domains of that graph. In this sense, we get the notion of a metric space of metric spaces where each of these spaces is represented by a separate generalised tree which satisfies certain structural constraints guaranteeing a functional semantics of its different edge types.

Remark. The notion of a conceptual graph is not to be confused with that of the same name as introduced by [54]. In contrast to the latter term, Definition 1.17 is reminiscent of the notion of a conceptual space as introduced by [27] by relying on the notion of a graph-like topology instead of referring to a hyperspace-based geometry. In Section 1.2.8 we have shown how to utilise generalised trees as graph models of conceptual domains. By Definition 1.17 we get the

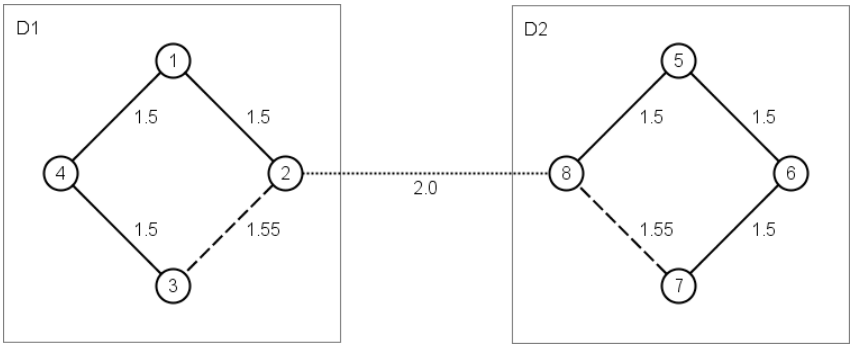

Fig. 1.8: A generalised forest with two components D_1 and D_2 (cf. the proof of Corollary 1.13). Kernel edges are denoted by straight lines, lateral edges by dashed lines and the single external edge by a dotted line. Numeric labels denote the weights of the edges.

understanding that these domains each of which is endowed with a graph-like topology, are interlinked by external edges which connect less coherent and, thus, separable vertices. In other words: Definition 1.17 reconstructs the opposition of separable and integral dimensions by the notion of micro and meso (level) coherence — leaving the definition of macro (level) coherence of conceptual graphs to future work: a domain is an internally structured, externally separable and internally integral dimension of a conceptual graph. We call a conceptual graph G for which $|\mathrm{cmp}(G)| \gg 1$ a *multi-dimensional conceptual space*. In Section 1.3 we exemplify this notion by three different semiotic systems.

Corollary 1.13 *A generalised forest can have micro level coherence without having meso level coherence.*

Proof. This Corollary can simply be proved by constructing a counterexample as shown in Figure 1.8 which is a generalised forest $G = (V, E^{\tau}_{[0..5]}, \mu)$ with two components $D_1 = (\{1,2,3,4\}, E^{\tau_1}_{[1..5]}, 1, \mu_1)$ and $D_2 = (\{5,6,7,8\}, E^{\tau_2}_{[1..5]}, 5, \mu_2)$ such that $E^{\tau}_{[0]} = \{\{2,8\}\}$, $E^{\tau_1}_{[2]} = E^{\tau_1}_{[3]} = E^{\tau_1}_{[4]} = E^{\tau_2}_{[2]} = E^{\tau_2}_{[3]} = E^{\tau_2}_{[4]} = \emptyset$, $E^{\tau_1}_{[1]} = \{\{1,2\},\{1,4\},\{3,4\}\}$, $E^{\tau_2}_{[1]} = \{\{5,6\},\{5,8\},\{6,7\}\}$, $E^{\tau_1}_{[5]} = \{\{2,3\}\}$ and $E^{\tau_2}_{[5]} = \{\{7,8\}\}$. Further, $\mu(\{1,2\}) = \mu(\{1,4\}) = \mu(\{3,4\}) = \mu(\{5,6\}) = \mu(\{5,8\}) = \mu(\{6,7\}) = 1.5$, $\mu(\{2,3\}) = \mu(\{7,8\}) = 1.55$, and $\mu(\{2,8\}) = 2.0$. Suppose now that $\mu^* = \hat{\mu}$. In this case we see that Constraint 1 of Definition 1.17 is satisfied while $\max\{\max_{x,y\in V_1}\{\mu^*(x,y)\}, \max_{x,y\in V_2}\{\mu^*(x,y)\}\} = 3.0 > 2.0 = \hat{\mu}(v,w)$. □

Obviously, Definition 1.17 demands spanning conceptual graphs in a sense that vertices of the same domain are "nearer" to each other, more related or

more similar than vertices of different domains. This is the generalised tree-based analogue to the distinction of integral (domain internal) and separable (domain external) dimensions in conceptual spaces. It reflects a basic idea of explorative data analysis according to which objects of the same cluster shall be more homogeneous than objects of different clusters — irrespective of the operative measure of object similarity. Although we do not specify μ^* in Definition 1.17, good candidates for instantiating μ^* can be derived from algorithms for clustering graphs (cf., e.g., [61]).

At this point we stop extending the graph-theoretical apparatus introduced so far and leave this endeavor for future work. What we finally present in the next Section is an overall interpretation of the notion of a generalised forest as introduced so far.

1.3 Semiotic Systems as Conceptual Graphs

So far we have gained several novel subclasses of the class of generalised trees. It was Dehmer's [15] task — who first and, up to that time, most comprehensively formalised GTs — to define a similarity measure for classifying *given* sets of generalised trees. That is, for a triple of generalised trees [15] determines the most similar pair of GTs. In this chapter we have made one step back in order to approach an answer to the question: *Given a single graph, which of the generalised trees derivable from it satisfies which topologically and semiotically founded constraints?* Following this line of research we have introduced the notion of a minimum spanning generalised tree (MSGT), of a generalised shortest path tree (GSPT) and of a generalised shortest path**s** tree (GPST). Especially by the subclass of GPSTs we have gained a detailed semantics of kernel, vertical, reflexive and lateral edges where the latter have further been divided into the subset of cross-reference and short cut edges. In Section 1.2.6 we have given a functional semantics of kernel edges as search facilities, of vertical edges as abridging facilities, of short cut edges as association facilities (in support of large cluster values) and of cross-reference edges as randomisation facilities (in support of short average geodesic distances). Generalised trees are a class of graphs which impose functional restrictions on the typing of their edges which locate this class in-between the class of trees and general graphs. In this sense, we have reached an information-added value from our new look on graphs: although each GT is a graph by definition, the latter semantics of edge types provides a detailed classification of edges according to their *function* in processes of information flow in networks. Further, in Section 1.2.8 and 1.2.9 we have approached the stage of elaborating the notion of a GT in terms of cognitive modeling. It is now that we explain this model by example of three semiotic domains.

In order to do that let us briefly recapitulate our instantiation of conceptual spaces by means of generalised forests as presented above: Starting from a graph we denote quality dimensions by its vertices whose networking into generalised trees defines domains, that is, internally structured dimensions of mutually integral basic dimensions. Starting from this setting, conceptual spaces are spanned by interlinked domains such that dimensions of different domains are less coherent than those belonging to the same domain (remember our analogue to the notion of separable dimensions). *So what does it mean to map an object on such a conceptual space?* There are at least two candidate answers to this question — a (neo-)structuralistic one and a conceptualistic one (as we call them):

1. *Structuralistic interpretation — the unipartite model:* According to this view, a conceptual space is a unipartite graph in which all entities — whether dimensions or objects — are mapped onto the same single mode of the graph. That way, any object is defined in accordance with the general stance of structuralism by its relative position with respect to all other objects of the same space [43] (in terms of direct or indirect links). Following this interpretation — which below is exemplified by text networks — a newly observed object o is mapped onto conceptual space

   a) by finding the domain to which it is best related (in terms of the operative notion of object relatedness or similarity),
   b) by locating o relative to the dimensions of that domain subject to the restrictions of generalised shortest paths trees and
   c) by establishing external edges which relate o to alternative domains as representations of its additional meanings.

   Under this regime, the root of a generalised shortest paths tree representing a certain domain is the prototype of this domain which by virtue of this structuralist interpretation is an existent sign [45] (and not just a virtual configuration of features). Further, domains (and the conceptual graphs spanned by them) necessarily grow as a function of newly made observations, that is, by newly made object measurements (e.g., processes of sign interpretation). This may also affect the rewiring of already established domain-internal or -external links. As a consequence, measurement operations are reconstructed as a sort of object wiring or edge formation. We can represent such kind of measurement operations by means of a vector-like notation as follows: an object o is mapped onto a conceptual graph G by a vector $\mathbf{o}^T = (d_1, \ldots, d_n)$ where $d_i = \mu(o, v_i)$, $1 \leq i \leq n$, iff o is linked with vertex v_i, otherwise $d_i = 0$. From this point of view, we naturally gain an interpretation of conceptual graphs accord-

CS	Conceptual Graph	Text Networking	Social Tagging	Thematic Progression
dimension	vertex	text	social category	text segment
domain	generalised tree	text subnetwork	social category subgraph	subnetwork of text segments
space	generalised forest	text network	social category graph	network of text segments
object	vertex of the same or different mode	text	text (segment)	text (segment)

Tab. 1.1: Overview of the notion of a conceptual graph as a purely graph-theoretical reconstruction of the notion of a *Conceptual Space* (CS) [27] and three of its instances.

ing to the theory of diachronic structuralism as pushed by [31], [13] and especially by [18].

2. *Conceptualistic interpretation — the bipartite model:* According to this view, a conceptual graph spans a bipartite graph with two modes: whereas the top mode is (as above) spanned by the interlinked domains of the graph, objects are now separately mapped onto the graph's bottom mode. There are two alternatives of representing this bottom mode: either objects are represented in the usual way as vectors of values along dimensions which span a geometric space or the object space spans itself a generalised forest. It is the latter variant which is preferred here. Following this interpretation, a conceptual space consists of two interrelated conceptual graphs, the one representing a system of dimensions, the other a system of objects characterised and interrelated along these dimensions. In order to make sense of this interpretation we are in need of a notion of commutativity by analogy to category theory. That is, links among dimensions restrict the set of links among objects. Under this regime, the root of a generalised shortest paths tree representing a certain domain is the prototype of this domain which by virtue of this conceptualistic interpretation is no longer a real existing sign [45].

Both of these interpretations naturally include hierarchical categorisation as a mode of object representation [**?**]. This is a direct consequence of the topology of generalised trees by which objects may be mapped onto vertices of different levels of the kernel of the operative domain. Note that both of these interpretations of conceptual spaces are contrary to feature semantics which (i) establishes *semes* [29] as semantic dimensions which form building blocks of (ii) categories as regions of semantic space onto which objects are (iii) mapped by categorising them along the latter categories. The reason why we do not follow this approach is the sheer impossibility of finding such reliable semantic dimensions. Thus, following the tradition of neo-structuralism we refer to the signs themselves as dimension building units [18].

Now we can exemplify the latter two interpretations of conceptual spaces in terms of conceptual graphs by means of three semiotic domains:

- *Text networking:* A first example of conceptual graphs can be constructed in the area of social text networking. The most prominent example of this is the Wikipedia in which articles and related document units are the vertices which are connected by encyclopedic links [40]. In this area we can build a conceptual graph along with a structuralist interpretation by identifying topical domains as sub-networks of Wikipedia's article graph. This can be done by clustering articles according to their content. However, Wikipedia also knows the concept of thematic portals which — as is easily shown [40] — have a generalised tree-like topology. Without elaborating this example in detail let us mention that while in this example kernel and vertical links express relations of thematic hypotaxis, subordination or containedness among articles, short cut edges connect articles related by textual entailment while cross-reference links can be used to represent links among thematically loosely connected articles (e.g. by means of dates, locations etc.). Following this line of thinking, the Wikipedia article graph gets a semantic space in itself which is subdivided into portals and other clusters of articles where each of these clusters spans a certain thematic domain by means of its thematic homogeneity. By mapping a single text (or a new article) onto this conceptual graph we get, amongst others, information about its membership to certain thematic domains. Additionally, for each of these domains representing its ambiguous content we get information about the degree of its thematic resolution (as a function of its geodesic distance to the root of that domain). Finally, by classifying all links starting from that text in terms of kernel, vertical, short cut and cross-reference edges we specify its functional position in processes of information flow through that network. This is just a structuralist way of revealing the content of an object by interlinking it with other objects of the same ontological sphere.
- *Social tagging:* Along with a conceptualistic reading of conceptual graphs we get a second example. Now instead of directly interrelating a text with the vertices of a given text network we can alternatively map that text onto the category system of the Wikipedia [63]. That way, the category system of the Wikipedia [62] is reconstructed as a generalised forest in which different subgraphs span thematically distinguished subject areas (e.g. *culture*, *science*, *sports*). That is, we assume that kernel edges model hypernymy relations while vertical edges abbreviate them according to their transitivity. Further, we assume that short cut edges map socially linked categories which denote co-classifications or polymorphic categorisations within a given thematic domain. Finally, cross-reference edges are seen to denote remote, that is, less obvious

co-classifications. Under this regime, external edges combine obviously unrelated domains of categorisation — possibly due to an erroneous co-classification of an ambiguous term or so. Obviously, vertical and lateral edges have quite different roles so that this information can be reflected by the built-up of generalised trees representing single domains of categorisations. Note that this interpretation in terms of interlinked domains relieves us of deciding on a top-level category. Such a single top-level category is as unrealistic as a purely tree-like skeleton of a category graph in social tagging. As before, when a text is mapped onto the resulting conceptual graph we perform a hierarchical categorisation where ambiguous texts are mapped onto different domains of the graph.

- *Thematic progression:* A third example belongs to the area of discourse analysis. According to the notion of thematic progression [14] we may think of a single discourse as being divided into interlinked thematic domains each of which represents a single topic separated from the rest of the topics of the same discourse. Representing these domains as generalised trees we decide to map thematic progressions by kernel edges which are supplemented by vertical edges as a means to abridge hypotactic relations among discontinuous text segments. Further, we can think of short cut edges as links among thematically associated text segments while cross-reference links denote thematically remote connections among randomly linked text parts. In other words, kernel and vertical edges model textual coherence as based on textual entailment, while short cut and cross-reference edges are used to model coherence relations of text segments based on thematic association. Note that the original model of thematic progression does not account for graph-like discourse structures but unrealistically relies on a tree-like model.

What have we gained by the graph-theoretical apparatus introduced so far? We have invented a graph model which shares the efficiency of shortest path trees with the expressiveness of graphs. Further, we have elaborated this notion along the notion of a conceptual space. More specifically, we have introduced generalised forests as a graph model which retains several non-trivial characteristics of semiotic systems. With respect to semantic relations this model accounts for

1. *thematic centralisation* according to the choice of prototypes as roots of generalised trees,
2. *hypotactic unfolding* by means of kernel edges along increasingly specialised nodes starting from the prototypical root of the generalised tree,
3. *thematic condensation* as provided by vertical links abridging taxonomical relations due to transitivity relations among kernel edges,

4. *thematic short cuts* as a means of representing thematic associations apart from taxonomic or otherwise hierarchical meaning relations,
5. *domain formation* as a result of networking among thematically homogeneous signs,
6. *domain networking* by spanning external edges among different domains in order to gain finally
7. *conceptual spaces* as reference systems of modeling the content of polysemous signs.

This reference example shows that generalised forests and their constitutive trees can be seen as a powerful tool for mapping semiotic systems of a wide range of areas (ranging from single texts and social ontologies to whole text networks). This opens the perspective on semiotic measurements beyond semantic spaces and the geometric model of meaning relations.

Acknowledgement

Financial support of the German Federal Ministry of Education (BMBF) through the research project *Linguistic Networks* and of the German Research Foundation (DFG) through the Excellence Cluster 277 *Cognitive Interaction Technology* (via the Project *Knowledge Enhanced Embodied Cognitive Interaction Technologies* (KnowCIT)), the SFB 673 *Alignment in Communication* (via the Project X1 *Multimodal Alignment Corpora: Statistical Modeling and Information Management*), the Research Group 437 *Text Technological Information Modeling* (via the Project A4 *Induction of Document Grammars for Webgenre Representation*) and the LIS-Project *Entwicklung, Erprobung und Evaluation eines Softwaresystems von inhaltsorientierten P2P-Agenten für die thematische Strukturierung und Suchoptimierung in digitalen Bibliotheken* at Goethe-University Frankfurt am Main and Bielefeld University, respectively, is gratefully acknowledged. We also thank Jolanta Bachan, Dafydd Gibbon and the anonymous reviewers for their fruitful hints which helped to reduce the number of errors in this chapter.

Bibliography

1 Albert, R. and Barabási, A.-L. (2002). Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74:47.

2 Baas, N. A. (1994). Emergence, hierarchies, and hyperstructures. In Langton, C. G., editor, *Artificial Life III, SFI Studies in the Sciences of Complexity*, pages 515–537. Addison-Wesley.

3 Barabási, A.-L. and Oltvai, Z. N. (2004). Network biology: Understanding the cell's functional organization. *Nature Reviews. Genetics*, 5(2):101–113.

4 Barrat, A., Barthélemy, M., Pastor-Satorras, R., and Vespignani, A. (2004). The architecture of complex weighted networks. *Proc. Nat. Acad. Sci. USA*, 101(11):3747–3752.

5 Barthélemy, M. (2004). Betweenness centrality in large complex networks. *European Physical Journal B*, 38:163–168.

6 Blanchard, P. and Krüger, T. (2004). The cameo principle and the origin of scale free graphs in social networks. *Journal of statistical physics*, 114(5-6):399–416.

7 Brainerd, B. (1977). Graphs, topology and text. *Poetics*, 1(14):1–14.

8 Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A., and Wiener, J. (2000). Graph structure in the web. *Computer Networks*, 33:309–320.

9 Budanitsky, A. and Hirst, G. (2006). Evaluating WordNet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47.

10 Burgess, C., Livesay, K., and Lund, K. (1999). Exploration in context space: Words, sentences, discourse. *Discourse Processes*, 25(2&3):211–257.

11 Chakrabarti, S. (2002). *Mining the Web: Discovering Knowledge from Hypertext Data*. Morgan Kaufmann, San Francisco.

12 Chazelle, B. (2000). A minimum spanning tree algorithm with inverse-ackermann type complexity. *JACM*, 47(6):1028–1047.

13 Coseriu, E. (1974). *Synchronie, Diachronie und Geschichte. Das Problem des Sprachwandels*. Fink, Wilhelm.

14 Daneš, F. (1974). Functional sentence perspective and the organization of the text. In Daneš, F., editor, *Papers on Functional Sentence Perspective*, pages 106–128. Mouton, The Hague.

15 Dehmer, M. (2005). *Strukturelle Analyse Web-basierter Dokumente*. Phd thesis, Technische Universität Darmstadt, Fachbereich Informatik, Berlin.

16 Dehmer, M. and Mehler, A. (2007). A new method of measuring the similarity for a special class of directed graphs. *Tatra Mountains Mathematical Publications*, 36:39–59.

17 Dehmer, M., Mehler, A., and Emmert-Streib, F. (2007). Graph-theoretical characterizations of generalized trees. In *Proceedings of the 2007 International Conference on Machine Learning: Models, Technologies & Applications (MLMTA'07), June 25-28, 2007, Las Vegas*, pages 113–117.

18 Derrida, J. (1988). *Limited Inc.* Northwestern University Press, Chicago.

19 Diestel, R. (2005). *Graph Theory*. Springer, Heidelberg.

20 Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numererische Mathematik*, 1:269–271.

21 Dorogovtsev, S. N. and Mendes, J. F. F. (2004). The shortest path to complex networks. http://www.citebase.org/abstract?id=oai:arXiv.org:cond-mat/0404593.

22 Ehresmann, A. C. and Vanbremeersch, J.-P. (1996). Multiplicity principle and ermegence in memory evolutive systems. *SAMS*, 26:81–117.

23 Emmert-Streib, F. and Dehmer, M. (2006). A systems biology approach for the classification of dna microarray data. In *Proceedings of ICANN 2005, Poland/Torun*.

24 Emmert-Streib, F. and Dehmer, M. (2007). Topological mappings between graphs, trees and generalized trees. *Applied Mathematics and Computing*, 186(2):1326–1333.

25 Emmert-Streib, F., Dehmer, M., and Kilian, J. (2005). Classification of large graphs by a local tree decomposition. In Arabnia, H. R. and Scime, A., editors, *Proceedings of DMIN'05, International Conference on Data Mining, Las Vegas, Juni 20-23*, pages 200–207.

26 Fischer, W. L. (1969). Texte als simpliziale Komplexe. *Beiträge zur Linguistik und Informationsverarbeitung*, 17:27–48.

27 Gärdenfors, P. (2000). *Conceptual Spaces*. MIT Press, Cambridge, MA.

28 Goldblatt, R. (1979). *Topoi: the Categorial Analysis of Logic*. Springer, Amsterdam.

29 Greimas, A. J. (2002). *Sémantique Structurale*. Presses Universitaires de France, Paris.

30 Gritzmann, P. (2007). On the mathematics of semantic spaces. In Mehler, A. and Köhler, R., editors, *Aspects of Automatic Text Analysis*, volume 209 of *Studies in Fuzziness and Soft Computing*, pages 95–115. Springer, Berlin/Heidelberg.

31 Jakobson, R. (1971). *Selected Writings II. Word and Language*. Mouton, The Hague.

32 Jones, W. and Furnas, G. (1987). Pictures of relevance: A geometric analysis of similarity measures. *Journal of the American Society for Information Science*, 38(6):420–442.

33 Kintsch, W. (1998). *Comprehension. A Paradigm for Cognition*. Cambridge University Press, Cambridge.

34 Landauer, T. K. and Dumais, S. T. (1997). A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104(2):211–240.

35 Lin, D. (1998). Automatic retrieval and clustering of similar words. In *Proceedings of the COLING-ACL '98*, pages 768–774.

36 Marcu, D. (2000). *The Theory and Practice of Discourse Parsing and Summarization*. MIT Press, Cambridge, Massachusetts.

37 Marcus, S. (1980). Textual cohesion and textual coherence. *Revue roumaine de linguistique*, 25(2):101–112.

38 Mehler, A. (2002). Hierarchical orderings of textual units. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING '02), August 24 – September 1, 2002, Taipei, Taiwan*, pages 646–652, San Francisco. Morgan Kaufmann.

39 Mehler, A. (2008a). Large text networks as an object of corpus linguistic studies. In Lüdeling, A. and Kytö, M., editors, *Corpus Linguistics. An International Handbook of the Science of Language and Society*, pages 328–382. De Gruyter, Berlin/New York.

40 Mehler, A. (2008b). Structural similarities of complex networks: A computational model by example of wiki graphs. *Applied Artificial Intelligence*, 22(7&8):619–683.

41 Mehler, A. (2009). Minimum spanning Markovian trees: Introducing context-sensitivity into the generation of spanning trees. In Dehmer, M., editor, *Structural Analysis of Complex Networks*. Birkhäuser Publishing, Basel.

42 Mehler, A. and Gleim, R. (2006). The net for the graphs – towards webgenre representation for corpus linguistic studies. In Baroni, M. and Bernardini, S., editors, *WaCky! Working Papers on the Web as Corpus*, pages 191–224. Gedit, Bologna.

43 Merleau-Ponty, M. (1993). *Die Prosa der Welt*. Fink, München.

44 Milgram, S. (1967). The small-world problem. *Psychology Today*, 2:60–67.

45 Murphy, G. L. (2002). *The big book of concepts*. MIT Press, Cambridge.

46 Newman, M. E. J. (2003). The structure and function of complex networks. *SIAM Review*, 45:167–256.

47 Pastor-Satorras, R. and Vespignani, A. (2004). *Evolution and Structure of the Internet*. Cambridge University Press, Cambridge.

48 Pickering, M. J. and Garrod, S. (2004). Toward a mechanistic psychology of dialogue. *Behavioral and Brain Sciences*, 27:169–226.

49 Rieger, B. B. (1978). Feasible fuzzy semantics. In *7th International Conference on Computational Linguistics (COLING-78)*, pages 41–43.

50 Schütze, H. (1997). *Ambiguity Resolution in Language Learning: Computational and Cognitive Models*, volume 71 of *CSLI Lecture Notes*. CSLI Publications, Stanford.

51 Serrano, M. Á., Boguñá, M., and Pastor-Satorras, R. (2006). Correlations in weighted networks. *Physical Review E*, 74:055101.

52 Shen-Orr, S., Milo, R., Mangan, S., and Alon, U. (2002). Network motifs in the transcriptional regulation network of escherichia coli. *Nature Genetics*, 31(1):64–68.

53 Smolensky, P. (1995). Connectionism, constituency and the language of thought. In Donald, M. and MacDonald, G., editors, *Connectionism: Debates on Psychological Explanation*, volume 2, pages 164–198. Blackwell, Oxford.

54 Sowa, J. F. (2000). *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Brooks/Cole, Pacific Grove.

55 Steels, L. (2006). Collaborative tagging as distributed cognition. *Pragmatics & Cognition*, 14(2):287–292.

56 Steyvers, M. and Tenenbaum, J. (2005). The large-scale structure of semantic networks: Statistical analyses and a model of semantic growth. *Cognitive Science*, 29(1):41–78.

57 Stroustrup, B. (2000). *Die C++-Programmiersprache*. Addison-Wesley, Bonn.

58 Tarjan, R. E. (1983). *Data structures and network algorithms*. Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania.

59 Thiopoulos, C. (1990). Meaning metamorphosis in the semiotic topos. *Theoretical Linguistics*, 16(2/3):255–274.

60 Tversky, A. and Gati, I. (2004). Studies of similarity. In Shafir, E., editor, *Preference, Belief, and Similarity. Selected Writing os Amos Tversky*, pages 75–95. MIT Press, Weinheim.

61 van Dongen, S. (2000). A cluster algorithm for graphs. Technical Report INS-R0010, National Research Institute for Mathematics and Computer Science in the Netherlands, Amsterdam.

62 Voss, J. (2006). Collaborative thesaurus tagging the Wikipedia way. arXiv.org:cs/0604036.

63 Waltinger, U., Mehler, A., and Heyer, G. (2008). Towards automatic content tagging: Enhanced web services in digital libraries using lexical chaining. In Cordeiro, J., Filipe, J., and Hammoudi, S., editors, *4th Int. Conf. on Web Information Systems and Technologies (WEBIST '08), 4-7 May, Funchal, Portugal*, pages 231–236. INSTICC Press, Barcelona.

64 Wasserman, S. and Faust, K. (1999). *Social Network Analysis. Methods and Applications*. Cambridge University Press, Cambridge.

65 Watts, D. J. (2003). *Six Degrees. The Science of a Connected Age*. W. W. Norton & Company, New York/London.

66 Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, 393:440–442.

67 Wu, B. Y. and Chao, K.-M. (2004). *Spanning Trees and Optimization Problems*. CRC Press, Boca Raton & London.

68 Yang, S. J. (2000). The directed minimum spanning tree problem. http://www.ce.rit.edu/~sjyeec/dmst.html.