Context Graphs for Many-Valued Contexts

Jens Kötters

Monash University, Melbourne, Australia

Abstract. A context graph of a one-valued context is a graph which has formal objects as its vertices, and whose edges connect the objects in such a way that concept extents form connected subgraphs of this graph. This allows to retrieve objects similar to a given object by a recursive traversal of that objects neighborhood. The approach has been introduced in [1] and is considered as a potential model for information retrieval.
In this paper, context graphs for many-valued contexts are introduced. The definition bases on descriptions, which were introduced in [2], and basically allows to construct context graphs without transforming the many-valued context into a one-valued context. It is examined how the structure of the context graph, which is based on similarity of objects, can be used to find objects in the graph based on a particular description. Finally, scale graphs are proposed as a method to incorporate knowledge about an attribute domain into the structure of the context graph.

Key words: Context Graphs, Formal Concept Analysis, Information Retrieval, Conceptual Scaling

1 Introduction

The concept lattices of Formal Concept Analysis (FCA) [3] have been studied and utilized as structural models for information retrieval (IR) by a number of researchers. The advantage concept lattices have to offer is that they support the combination of two principal ways of user interaction with an information retrieval system: querying and browsing. The most basic idea of such an application, described e.g. in [4], is that a subset of the attributes of a formal context is understood as a query by the system. The corresponding result set is the concept extent generated by the attributes in the query. This way, every query positions the user in a concept node of the lattices Hasse diagram, and the system can then present an option to the user to move to a lower or upper neighbor of the current node. Such a step in the diagram corresponds to a minimal refinement or extension of the result set, and this is how browsing is realized.

The complexity of the Hasse diagram sets a natural limit to the size of formal contexts for which the Hasse diagram can be realistically computed. More recent publications present ways how these limits can be advanced, in addition to presenting more sophisticated user interfaces [5–7]. In [1] we have introduced a graph representation of a formal context that we call a context graph, and have motivated them as a potential alternative to the concept lattice in IR applications and beyond. The nodes of a context graph are objects of a formal context, and these are connected by edges in such a way that every concept extent is a connected subgraph within the context graph. We will generally assume that the graph has no more edges than necessary for this property to hold. The rationale of this approach is that, in an IR application, as soon as *one* object matching a given query is found, all other matches can be retrieved by recursively traversing that objects neighborhood. Moreover, the conceptual hierarchy of the lattice is contained in the context graph as a system of subgraphs, which means that navigating in the lattice (aka "browsing") can be simulated in the graph (although some search may be involved). An example context graph for the "Living Beings and Water" context (Fig. 3) is shown on the left of Fig. 4.

So far context graphs and the algorithm for their construction in [1] have only been defined for contexts with one-valued attributes. In a practical scenario however, objects that one might want to find in an IR system are more likely to be described by many-valued attributes (e.g. price or location). In FCA, the classic approach of dealing with many-valued attributes is to translate a many-valued context into a one-valued context by means of conceptual scaling [3], but other ways of generating a lattice from a many-valued context have been proposed (see e.g. [2, 8]). In this paper we will follow the approach of Gugisch using descriptions [2]. In Sect. 3, the definition of context graphs and the pivotal notion of a compliant path are reformulated in terms of descriptions and we show that the resulting context graphs are equivalent to what we would obtain by conceptual scales. Thus we provide a true generalization of context graphs into the setting of many-valued attributes. An adaptation of the construction algorithm from [1] is straightforward and will not be discussed here.

A context graph could be likened to a kind of semantic map, where objects are distributed according to their similarity. This leads to believe that it may be possible to find an object in the graph that matches a given description by starting in some arbitrary vertex and following along a path of subsequently better matches. This will be examined in Sect. 4, using the notions developed in Sect. 3.

Finally, in Sect. 5 we briefly present the idea of scale graphs. Scale graphs are context graphs of a scale and therefore express similarity between attribute values. The proposed idea is that attribute domains can be intuitively modeled by drawing a graph and then defining an appropriate scale which has this graph as a context graph. The scale graph will then be reflected in the structure of any context graph created with the corresponding scale.

2 Basic Notions

2.1 Formal Concept Analysis

A *formal context* is a triple (G, M, I) consisting of two sets G and M and a relation $I \subseteq G \times M$. The members of G and M are called *objects* and *attributes*, respectively. We say that an object $g \in G$ has an attribute $m \in M$ if $(g, m) \in I$.

Let $\mathbb{K} := (G, M, I)$ be a formal context. The set of attributes of an object $g \in G$ is

$$\operatorname{att}_{\mathbb{K}}(g) := \{m \in M \mid gIm\} \ . \tag{1}$$

We extend the definition to sets of objects: The set of attributes shared by all objects of a subset $A \subseteq G$ is

$$\operatorname{att}_{\mathbb{K}}(A) := \bigcap_{g \in A} \operatorname{att}_{\mathbb{K}}(g) \ . \tag{2}$$

The set of objects which have all attributes of a subset $B \subseteq M$ is

$$\operatorname{obj}_{\mathbb{K}}(B) := \{g \in G \mid B \subseteq \operatorname{att}_{\mathbb{K}}(g)\} \ . \tag{3}$$

A pair (A, B) with $A \subseteq G$ and $B \subseteq M$ is called a *formal concept* of $\mathbb{K}$ if $A = \operatorname{obj}_{\mathbb{K}}(B)$ and $B = \operatorname{att}_{\mathbb{K}}(A)$. In this case A is called the *extent* and B the *intent* of the formal concept (A, B). We write $\mathcal{B}(\mathbb{K})$ for the set of all formal concepts of $\mathbb{K}$.

The smallest concept extent which contains a set $A \subseteq G$ of objects is given by

$$\overline{A}^{\mathbb{K}} := \operatorname{obj}_{\mathbb{K}}(\operatorname{att}_{\mathbb{K}}(A)) \text{ for } A \subseteq G \ . \tag{4}$$

We will also use the abbreviations $\overline{gh}^{\mathbb{K}} := \overline{\{g, h\}}^{\mathbb{K}}$ and $\overline{gA}^{\mathbb{K}} := \overline{\{g\} \cup A}^{\mathbb{K}}$ for $g, h \in G$ and $A \subseteq G$. The smallest concept intent which contains a set $B \subseteq M$ of attributes is given by

$$\overline{B}^{\mathbb{K}} := \operatorname{att}_{\mathbb{K}}(\operatorname{obj}_{\mathbb{K}}(B)) \text{ for } B \subseteq M \ . \tag{5}$$

We can characterize the set of all concepts in the following ways:

$$\mathcal{B}(\mathbb{K}) = \{(\overline{A}^{\mathbb{K}}, \operatorname{att}_{\mathbb{K}}(A)) \mid A \subseteq G\} \ , \tag{6}$$

$$\mathcal{B}(\mathbb{K}) = \{(\operatorname{obj}_{\mathbb{K}}(B), \overline{B}^{\mathbb{K}}) \mid B \subseteq M\} \ . \tag{7}$$

The extents and intents of two concepts $(A^{(1)}, B^{(1)})$ and $(A^{(2)}, B^{(2)})$ of $\mathbb{K}$ are related by

$$A^{(1)} \subseteq A^{(2)} \Leftrightarrow B^{(1)} \supseteq B^{(2)} \ . \tag{8}$$

In the case that $A^{(1)} \subseteq A^{(2)}$ holds, we call $(A^{(1)}, B^{(1)})$ a *subconcept* of $(A^{(2)}, B^{(2)})$ and write $(A^{(1)}, B^{(1)}) \leq (A^{(2)}, B^{(2)})$. The partially ordered set $(\mathcal{B}(\mathbb{K}), \leq)$ is a complete lattice, called the *concept lattice* of $\mathbb{K}$.

2.2 Graph Theory

An *undirected graph* is a pair $K = (V_K, E_K)$, where V_K is a finite set and E_K is a subset of $\{\{x, y\} \subseteq V_K \mid x \neq y\}$. The members of V_K are called the *vertices*, and the members of E_K the *edges* of K. Given two vertices $x, y \in V_K$, we say that x and y are joined by an edge if $\{x, y\} \in E_K$. In this case, we call x and y *neighbors* and we may alternatively write $x \sim y$.

A sequence $(v^{(1)}, \dots, v^{(k)})$ of vertices is called a *walk* from $v^{(1)}$ to $v^{(k)}$ if $v^{(i)} \sim v^{(i+1)}$ for $i = 1, \dots, k-1$. If no two vertices on the walk are the same, except possibly $v^{(1)} = v^{(k)}$, the walk is called a *path*. A path with $v^{(1)} = v^{(k)}$ is called a *circle*. We say that K is *connected* if there is a path from x to y for all $x, y \in V_K$.

A graph (V, E) with $V \subseteq V_K$ and $E \subseteq E_K$ is called a *subgraph* of K. The *induced subgraph* on a set $V \subseteq V_K$ of vertices is the subgraph $K[V] := (V, E_K \cap \mathcal{P}(V))$, where $\mathcal{P}(V)$ denotes the power set of V.

2.3 Context Graphs

Definition 1 (Context Graph of a Formal Context). *A* context graph *of a formal context $\mathbb{K} = (G, M, I)$ is a triple (G, E, f) where*

(CG1) *$K := (G, E)$ is an undirected graph on the set of objects,*
(CG2) *E is an arbitrary set of edges such that every induced subgraph $K[\mathrm{obj}_{\mathbb{K}}(B)]$, $B \subseteq M$, is connected,*
(CG3) *f is a labeling function on the vertex set with $f(g) = \mathrm{att}_{\mathbb{K}}(g)$ for $g \in G$.*

It has been shown in [1] how a context graph with a minimum number of edges can be constructed.

Definition 2 (Compliant Path). *A path $(x^{(1)}, \dots, x^{(k)})$ in a context graph is called* compliant *if $x^{(i)} \in \overline{x^{(1)}x^{(k)}}^{\mathbb{K}}$ for all $i = 1, \dots, k$.*

Condition (CG2) for context graphs is equivalent to the statement that there exists a compliant path between every two vertices $x, y \in G$. This follows from (6), (7) and, for all $A \subseteq G$,

$$\overline{A}^{\mathbb{K}} = \bigcup_{x,y \in \overline{A}^{\mathbb{K}}} \overline{xy}^{\mathbb{K}} \ . \tag{9}$$

This has already been shown in [1], where we have used a slightly different, but equivalent definition of compliant paths.

3 Context Graphs for Many-Valued Contexts

3.1 Conceptual Scaling

We use attributes like e.g. size, price, date or color to describe the objects that surround us. It is usually the value of such an attribute that we are interested in, not just the fact that the attribute applies to an object. We will call these attributes *many-valued attributes*, in contrast to *one-valued attributes* which simply do or do not apply to an object.

If we want to use Formal Concept Analysis on objects with many-valued attributes, we have to translate the many-valued attributes into one-valued attributes. As an example, “is big”, “is cheap”, “before yesterday” and “is black” may be some of the one-valued attributes which are used instead of size, price, date and color. This process is called *conceptual scaling*, and we will cover in this article the simplest variant, which is also described in [3].

The first step is to select a formal context called a *scale* for each attribute. The objects of the scale are the possible values of the many-valued attribute. We can consider the extents of the scale to be the meaningful subsets of values. For example, consider an attribute "rating" with possible values in $\{1 = \mathit{excellent}, 2 = \mathit{good}, 3 = \mathit{average}, 4 = \mathit{bad}\}$. This could be something like the user ratings for books and compact disks that you can find on Amazon.com, for example. We reason that this attribute could be scaled by the ordinal scale in Fig. 1:

	≤ 1	≤ 2	≤ 3	≥ 2	≥ 3	≥ 4
1	×	×	×			
2		×	×	×		
3			×	×	×	
4				×	×	×

	≤ 1	≤ 2	≤ 3
1	×	×	×
2		×	×
3			×
4			

	true	false
true	×	
false		×

Fig. 1. Interval Scale, Ordinal Scale and Dichotomic Scale

The extents of the ordinal scale are $\{1\}$, $\{1, 2\}$, $\{1, 2, 3\}$ and $\{1, 2, 3, 4\}$. If someone is interested in an item that is rated good, they would probably also accept an excellent item, so the extents of the ordinary scale are possible descriptions of what one might be looking for. The one-valued attributes of the scale are used to identify these subsets of values. In our example, one attribute is sufficient for each of the sets (e.g. "≤ 3" for $\{1, 2, 3\}$).

There are situations where somebody may be interested in bad items exclusively. The interval scale (see also Fig. 1) provides a more fine-grained description of integer value sets: The extents are all intervals $[a, b]$ with $a, b \in \{1, 2, 3, 4\}$, including the interval $[4, 4]$ that we just mentioned. To sum up, we can say that the scale chosen for an attribute determines which subsets of values are considered for the formation of concepts.

Figure 2 shows an example of how objects with many-valued attributes can be represented in a formal context. In this example, we have only three objects o_1, o_2 and o_3 which are described by a 5-tuple of attribute values each. We have scaled the first three attributes using the interval scale, dichotomic scale and ordinal scale of Fig. 1, respectively. The fourth attribute has not been scaled at all; this is possible as it only takes the values "true" or "false". The last attribute has been scaled by the color scale in Fig. 6.

The formal context obtained in this way is called the *derived context w.r.t. plain scaling*. We will formalize the process of conceptual scaling in the next section, taking a slightly different approach than that in [3].

Derived Context Example	≤ 1	≤ 2	≤ 3	≥ 2	≥ 3	≥ 4	true	false	≤ 1	≤ 2	≤ 3	true	r-o-y	o-y-g	y-g-b	g-b-p	b-p-r	p-r-o
o_1:(2,true,3,false,blue)		×	×	×			×				×				×	×	×	
o_2:(1,true,4,true,red)	×	×	×				×					×	×				×	×
o_3:(4,false,1,false,yellow)				×	×	×		×	×	×	×		×	×	×			

Fig. 2. Derived Context Example

3.2 Precontexts

We start with the definition of a precontext, which is a formal representation of the set of many-valued attributes that we initially have:

Definition 3 (Precontext). *A* precontext *is a tuple* $(G, W_1, \dots, W_n)$ *consisting of* $n \geq 1$ *nonempty sets* $W_1, \dots, W_n$ *and a set* $G \subseteq W_1 \times \dots \times W_n$*. We call a set* W_i *an* attribute domain*, and the members of* W_i *are called* values*. The members of* G *are called* objects*, and we say that an object* $(x_1, \dots, x_n) \in G$ *has a value of* $w \in W_i$ *in an attribute* $i \in \{1, \dots, n\}$ *if and only if* $w = x_i$*.*

Precontexts are less general than many-valued contexts (see [3]) in two respects:

1. The condition $G \subseteq W_1 \times \dots \times W_n$ means that objects are identified with the tuple of their attribute values. That is, objects with identical values in each attribute are considered the same.
2. The attributes are modeled by the projection functions π_i, $i = 1, \dots, n$. That is, the i-th attribute of an object $x = (x_1, \dots, x_n)$ is given by $\pi_i(x) = x_i$. In particular, the attributes are total functions, whereas the attributes of a many-valued context may be partial functions.

The first of these points is not an actual restriction if we want to navigate in context graphs, since objects with identical descriptions can be represented by the same vertex. The second point *is* a restriction: The current paper does not deal with the case where attribute values are unknown or undefined.

With this in mind, we can understand a precontext $(G, W_1, \dots, W_n)$ as an alternative representation for the many-valued context $(G, \{\pi_i \mid 1 \leq i \leq n\}, \bigcup_{i=1}^{n} W_i, I)$ with $I = \{(x, \pi_i, \pi_i(x)) \mid x \in G, 1 \leq i \leq n\}$.

Definition 4 (Precontext with Scales). *A* precontext with scales *is a tuple* $(G, \mathbb{S}_1, \dots, \mathbb{S}_n)$ *such that every* $\mathbb{S}_i$ *is a formal context* (W_i, M_i, I_i)*, and* $(G, W_1, \dots, W_n)$ *is a precontext. The contexts* $\mathbb{S}_i$ *are called* scales*.*

A precontext with scales $(G, \mathbb{S}_1, \dots, \mathbb{S}_n)$, with $\mathbb{S}_i = (W_i, M_i, I_i)$ for $i = 1, \dots, n$, and the corresponding derived context $\mathbb{K} = (G, M, I)$ are related by the equations

$$M = M_1 \mathbin{\dot\cup} \dots \mathbin{\dot\cup} M_n \ , \tag{10}$$

$$\forall x \in G : \mathrm{att}_{\mathbb{K}}(x) \cap M_i = \mathrm{att}_{\mathbb{S}_i}(\pi_i(x)) \ . \tag{11}$$

3.3 Descriptions

Theorem 1. *Let $(G, \vec{\mathbb{S}})$ be a precontext with scales and $\mathbb{K}$ the derived context w.r.t. plain scaling. Then*

$$\overline{A}^{\mathbb{K}} = G \cap \bigtimes_{i=1}^{n} \overline{\pi_i(A)}^{\mathbb{S}_i}$$

holds for all $A \subseteq G$.

Proof. We first note that

$$\begin{aligned} \mathrm{att}_{\mathbb{K}}(A) \cap M_i &\underset{(2)}{=} \bigcap_{x \in A} \mathrm{att}_{\mathbb{K}}(x) \cap M_i \underset{(11)}{=} \bigcap_{x \in A} \mathrm{att}_{\mathbb{S}_i}(\pi_i(x)) \\ &\underset{(2)}{=} \mathrm{att}_{\mathbb{S}_i}(\bigcup_{x \in A} \pi_i(x)) = \mathrm{att}_{\mathbb{S}_i}(\pi_i(A)) \ . \end{aligned} \tag{12}$$

From this we conclude

$$\begin{aligned} &\mathrm{att}_{\mathbb{K}}(A) \subseteq \mathrm{att}_{\mathbb{K}}(x) \\ \underset{(10)}{\Leftrightarrow} \ &\forall 1 \leq i \leq n : \mathrm{att}_{\mathbb{K}}(A) \cap M_i \subseteq \mathrm{att}_{\mathbb{K}}(x) \cap M_i \\ \underset{(12)(11)}{\Leftrightarrow} \ &\forall 1 \leq i \leq n : \mathrm{att}_{\mathbb{S}_i}(\pi_i(A)) \subseteq \mathrm{att}_{\mathbb{S}_i}(x_i) \\ \underset{(4)(3)}{\Leftrightarrow} \ &\forall 1 \leq i \leq n : x_i \in \overline{\pi_i(A)}^{\mathbb{S}_i} \ . \end{aligned} \tag{13}$$

Finally we have

$$\overline{A}^{\mathbb{K}} \underset{(4)(3)}{=} \{x \in G \mid \mathrm{att}_{\mathbb{K}}(A) \subseteq \mathrm{att}_{\mathbb{K}}(x)\} \underset{(13)}{=} G \cap \bigtimes_{i=1}^{n} \overline{\pi_i(A)}^{\mathbb{S}_i} \ .$$

□

The theorem provides a description of the extents of the derived context as products of value sets. We formalize this notion of description in the following definition (descriptions have been explored in [2] already, although the definition given there is not exactly the same).

Definition 5 (Description). *Let $\mathbb{M}$ be a precontext with scales $\mathbb{S}_i = (W_i, M_i, I_i)$, $i = 1, \ldots, n$. A* description *in $\mathbb{M}$ is a set $D = D_1 \times \cdots \times D_n \subseteq W_1 \times \cdots \times W_n$ such that D_i is an extent of $\mathbb{S}_i$ for all $i = 1, \ldots, n$. We write $\mathcal{D}(\mathbb{M})$ for the set of all descriptions in $\mathbb{M}$.*

An object description *is a description $D \in \mathcal{D}(\mathbb{M})$ with $D_i = \overline{h_i}^{\mathbb{S}_i}$ for some $h \in W_1 \times \cdots \times W_n$, and we shall also write $D = h$ for convenience.*

In analogy to (2), (3), (4) and (5) we define

$$\mathrm{dsc}_{\mathbb{M}}(A) := \bigtimes_{i=1}^{n} \overline{\pi_i(A)}^{\mathbb{S}_i} \ , \tag{14}$$

$$\mathrm{obj}_{\mathbb{M}}(D) := G \cap D \ , \tag{15}$$

$$\overline{A}^{\mathbb{M}} := \mathrm{obj}_{\mathbb{M}}(\mathrm{dsc}_{\mathbb{M}}(A)) \ , \tag{16}$$

$$\overline{D}^{\mathbb{M}} := \mathrm{dsc}_{\mathbb{M}}(\mathrm{obj}_{\mathbb{M}}(D)) \ . \tag{17}$$

for $A \subseteq G$ and $D \in \mathcal{D}(\mathbb{M})$. Now we can state Theorem 1 in the shorter form

$$\overline{A}^{\mathbb{K}} = \overline{A}^{\mathbb{M}} \ . \tag{18}$$

We write $\mathcal{B}(\mathbb{M})$ for the set of all pairs (A, D) with $A = \mathrm{obj}_{\mathbb{M}}(D)$ and $D = \mathrm{dsc}_{\mathbb{M}}(A)$ and state without proof that

$$\mathcal{B}(\mathbb{M}) = \{(\overline{A}^{\mathbb{M}}, \mathrm{dsc}_{\mathbb{M}}(A)) \mid A \subseteq G\} \ , \tag{19}$$

$$\mathcal{B}(\mathbb{M}) = \{(\mathrm{obj}_{\mathbb{M}}(D), \overline{D}^{\mathbb{M}}) \mid D \in \mathcal{D}(\mathbb{M})\} \ . \tag{20}$$

Note that $A \subseteq \overline{A}^{\mathbb{M}}$, but $D \supseteq \overline{D}^{\mathbb{M}}$. Moreover,

$$A^{(1)} \subseteq A^{(2)} \Leftrightarrow D^{(1)} \subseteq D^{(2)} \tag{21}$$

for all $(A^{(1)}, D^{(1)}), (A^{(2)}, D^{(2)}) \in \mathcal{B}(\mathbb{M})$, which is different from what we have in (8).

3.4 The Context Graph of a Precontext with Scales

In this section we define the context graph for a precontext with scales:

Definition 6 (Context Graph of a Precontext with Scales). *Let $\mathbb{M} = (G, \vec{\mathbb{S}})$ be a precontext with scales. An undirected graph $K = (G, E)$ is called a* context graph *of $\mathbb{M}$, if $K[\mathrm{obj}_{\mathbb{M}}(D)]$ is connected for all descriptions D in $\mathbb{M}$.*

The sets $\mathrm{obj}_{\mathbb{M}}(D)$, $D \in \mathcal{D}(\mathbb{M})$, are precisely the extents $\overline{A}^{\mathbb{M}}$ ((19) and (20)). Because of (18) this means that we essentially get the same context graphs that we would get by first creating the derived context $\mathbb{K}$ and then using Definition 1. However, the attributes of an object do not have to be explicitly encoded by one-valued attributes, which is an advantage of this definition.

As in Sect. 2.3 we call a path between $x, y \in G$ compliant if it lies in $\overline{xy}^{\mathbb{M}}$ and obtain the following characterization:

Lemma 1. *Let $\mathbb{M} = (G, \mathbb{S}_1, \ldots, \mathbb{S}_n)$ be a precontext with scales. An undirected graph $K = (G, E)$ is a context graph of $\mathbb{M}$ if and only if there is a compliant path between all $x, y \in G$. A path $(x, \ldots, y)$ is compliant if and only if*

$$\forall i : z_i \in \overline{x_i y_i}^{\mathbb{S}_i}$$

holds for all z on the path.

4 Approximating Objects by Description

The following definition captures the idea of how close a vertex x matches a given description $D = D_1 \times \cdots \times D_n$:

$$\vec{u}_D(x) = (\overline{x_1 D_1}^{\mathbb{S}_1}, \ldots, \overline{x_n D_n}^{\mathbb{S}_n}) \ . \tag{22}$$

The vector contains, in every component, the smallest scale extent that contains $\{x_i\} \cup D_i$. This could be compared with a neighborhood of the set D_i in topology.

We write $\vec{u}_D(x) \leq \vec{u}_D(y)$ if x matches D better than y does:

$$\vec{u}_D(x) \leq \vec{u}_D(y) :\Leftrightarrow \forall i : \overline{x_i D_i}^{\mathbb{S}_i} \subseteq \overline{y_i D_i}^{\mathbb{S}_i} \ . \tag{23}$$

Definition 7 (Nonincreasing/Decreasing Paths). *Let K be a context graph of $\mathbb{M}$ and $D \in \mathcal{D}(\mathbb{M})$. A path $(x^{(1)}, \ldots, x^{(k)})$ is* nonincreasing *w.r.t. D if*

$$\vec{u}_D(x^{(1)}) \geq \cdots \geq \vec{u}_D(x^{(k)}) \ .$$

The path is decreasing *w.r.t. D if*

$$\vec{u}_D(x^{(1)}) > \cdots > \vec{u}_D(x^{(k)}) \ .$$

For a given description $D \in \mathcal{D}(\mathbb{M})$ we are interested in the objects which best match the description, i.e. the vertices $y \in G$ for which $\vec{u}_D(y)$ is minimal. We will see that from every vertex $x \in G$ there exists a nonincreasing path to some minimum. We will also prove conditions under which there exist nonincreasing paths to *all* minima beneath x, and conditions under which these nonincreasing paths are actually decreasing. The conditions depend on the scales chosen for the precontext.

Definition 8 (Scale Properties). *The following are properties which may be true or not for a given scale $\mathbb{S} = (W, M, I)$:*

(P1) $w \in \overline{uT}^{\mathbb{S}}, v \in \overline{uw}^{\mathbb{S}} \Rightarrow w \in \overline{vT}^{\mathbb{S}}$ *for all $u, v, w \in W$ and $T \subseteq W$*
(P2) $v \in \overline{uw}^{\mathbb{S}}, v \neq u, v \neq w \Rightarrow \overline{vw}^{\mathbb{S}} \subset \overline{uw}^{\mathbb{S}}$ *for all $u, v, w \in W$*

The given properties express what one would probably expect if the scale extents are seen as some kind of intervals, where $\overline{uw}^{\mathbb{S}}$ is the set of all values "between" u and w. Property (P2) then says that the interval $\overline{vw}$ generated by the inner point v and the end point w is strictly smaller. In the same fashion (P1) would mean that whenever w is between u and T and v is between u and w, then w must be between v and T.

The following lemma will give first results:

Lemma 2. *Let (G, E) be a context graph of $\mathbb{M}$, $D \in \mathcal{D}(\mathbb{M})$ and $x, y \in G$. Let further $(x, \ldots, y)$ be a compliant path between x and y. The following holds for all z on the path:*

1. $\vec{u}_D(x) \geq \vec{u}_D(z)$,

2. If all scales of $\mathbb{M}$ satisfy (P1): $\vec{u}_D(x) \geq \vec{u}_D(z) \geq \vec{u}_D(y)$.

Proof. Because z lies on a compliant path between x and y, we obtain

$$\forall i : z_i \in \overline{x_i y_i}^{\mathbb{S}_i} \tag{24}$$

from Lemma 1. From $\vec{u}_D(x) > \vec{u}_D(y)$ we conclude (starting with (23))

$$\begin{aligned}\overline{y_i D_i}^{\mathbb{S}_i} \subseteq \overline{x_i D_i}^{\mathbb{S}_i} &\Rightarrow y_i \in \overline{x_i D_i}^{\mathbb{S}_i} \Rightarrow \overline{x_i y_i}^{\mathbb{S}_i} \subseteq \overline{x_i D_i}^{\mathbb{S}_i} \\ &\underset{(24)}{\Rightarrow} z_i \in \overline{x_i D_i}^{\mathbb{S}_i} \Rightarrow \overline{z_i D_i}^{\mathbb{S}_i} \subseteq \overline{x_i D_i}^{\mathbb{S}_i} \ .\end{aligned}$$

The last inclusion shows that $\vec{u}_D(x) \geq \vec{u}_D(z)$. If all scales of $\mathbb{M}$ satisfy (P1), we obtain $\vec{u}_D(z) \geq \vec{u}_D(y)$ by setting $u := x_i$, $w := y_i$, $v := z_i$ and $T := D_i$ in Definition 8. □

Now let us have a closer look at the Lemma. We interpret x as an arbitrary vertex and y as a minimum beneath x. In a context graph there must be a compliant path p from x to y, and we write $p = (z^{(1)}, \ldots, z^{(m)})$ where $z^{(1)} = x$ and $z^{(m)} = y$. We follow this path until we either reach y or until we arrive at the first vertex $z^{(i)}$ with $\vec{u}_D(z^{(i-1)}) \neq \vec{u}_D(z^{(i)})$, i.e. with $\vec{u}_D(z^{(i-1)}) > \vec{u}_D(z^{(i)})$. If (P2) holds for all scales, then the Lemma says that y is still beneath $z^{(i)}$, and again there must be a complaint path from $z^{(i)}$ to y, which is not necessarily the remainder of the first path. But we can repeat the procedure with $z^{(i)}$ as the start vertex, approximating y step by step. That is, if (P2) holds for all scales, every minimum beneath x can be reached via a nonincreasing path. If (P2) does not hold, then y does not necessarily lie beneath $z^{(i)}$ anymore, but there will obviously still be some other minimum beneath $z^{(i)}$. We illustrate the latter case by an example:

The left side of Fig. 4 shows a context graph of the one-valued context in Fig. 3. Every one-valued context can be represented as a precontext with scales, using a scale $\mathbb{S}_{\text{id}}(\{0,1\},\{a\},\{(1,a)\})$ for each attribute a. This is easily verified showing that the derived context, defined by (10) and (11), is the original context again. The objects are then encoded by n-tuples of attributes in the obvious way (consider the right graph in Fig. 4 for this). The corresponding precontext with scales is therefore $(\{0,1\}^n, \mathbb{S}_{\text{id}}, \ldots, \mathbb{S}_{\text{id}})$. The extents of $\mathbb{S}_{\text{id}}$ are $\{0,1\}$ and $\{1\}$. We now want to find a best match for the description $D = (\{0,1\},\{0,1\},\{1\},\{0,1\},\{1\},\{0,1\},\{0,1\},\{1\},\{1\})$, which happens to be an object description (the corresponding object is $(0,0,1,0,1,0,0,1,1)$, which does not exist in the context). We start the search in the "Leech" vertex, which is represented by the tuple $(1,1,0,0,0,0,1,0,0)$. We could now use the definition in (23) to describe how closely the current vertex matches the description, but it is clear that we can equivalently use the set of shared attributes as a measure instead (the unmodified definition of $\vec{u}$ is better reserved for theoretical considerations). The attributes of the leech are given by $\{a,b,g\}$, and the object description translates to the set $\{c,e,h,i\}$. There are two best matches in the graph: the dog vertex and the bean vertex. Both lie below the leech vertex, and both can be reached via exactly one compliant path. But in the frog vertex, which lies two vertices down the path, the bean vertex does no longer lie beneath the frog vertex. So only the dog vertex can be reached via a nonincreasing path. In fact it can be seen that $\mathbb{S}_{\mathrm{id}}$ does not satisfy (P1).

Living Beings and Water	a	b	c	d	e	f	g	h	i
Leech	×	×					×		
Bream	×	×					×	×	
Frog	×	×	×				×	×	
Dog	×		×				×	×	×
Spike–weed	×	×		×		×			
Reed	×	×	×	×		×			
Bean	×		×	×	×				
Maize	×		×	×		×			

Fig. 3. Formal Context: Living Beings and Water (taken from [3]). The attributes are defined as follows: a=needs water, b=lives in water, c=lives on land, d=needs chlorophyll, e=two seed leaves, f=one seed leaf, g=can move, h=has limbs, i=suckles its offspring.

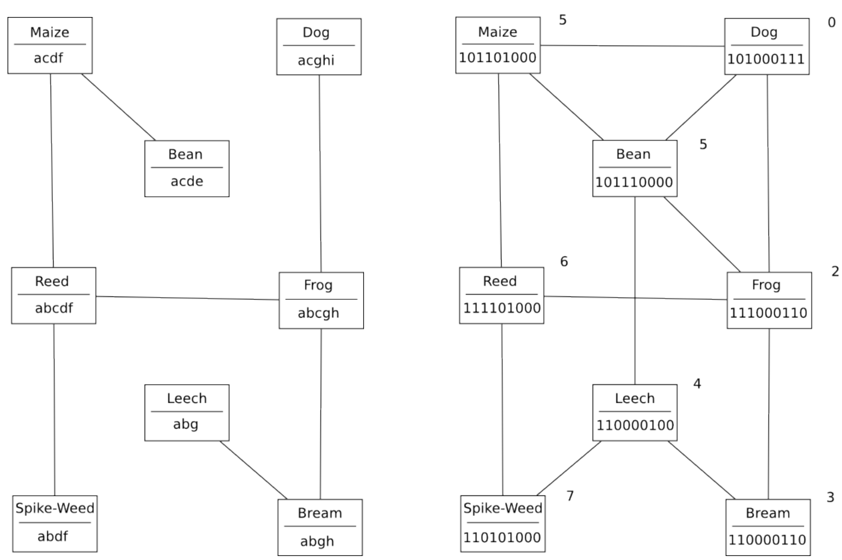


Fig. 4. Living Beings and Water: Context Graphs

The next lemma states that every nonincreasing path is in fact a decreasing path if all scales satisfy (P2) and the description is an object description.

Lemma 3. *Let (G, E) be a context graph of a precontext with scales $\mathbb{M}$. If all scales satisfy (P2), then*

$$x \neq y \Rightarrow \vec{u}_h(x) \neq \vec{u}_h(y)$$

holds for all $x, y \in G$ and all object descriptions $h \in \mathcal{D}(\mathbb{M})$.

Proof. Suppose that $x \neq y$ and $\vec{u}_h(x) = \vec{u}_h(y)$. Then for some $i \in \{1, \ldots, n\}$ we have $x_i \neq y_i$, wlog $x_i \neq h_i$, and $\overline{x_i h_i}^{\mathbb{S}_i} = \overline{y_i h_i}^{\mathbb{S}_i}$. The latter implies $x_i \in \overline{y_i h_i}^{\mathbb{S}_i}$, and from (P2) we obtain (setting $v := x_i$, $u := y_i$ and $w := h_i$) that $\overline{x_i h_i}^{\mathbb{S}_i} \subset \overline{y_i h_i}^{\mathbb{S}_i}$, contradiction! $\square$

The property (P2) is satisfied by the dichotomic scale and the interval scale. The right of Fig. 4 shows a context graph for the "Living Beings and Water" context where the precontext with scales is as above, but the scales $\mathbb{S}_{\text{id}}$ have been replaced by dichotomic scales. The corresponding derived context is shown in Fig. 5. Consider as a second example that we want to search this graph for

Living Beings and Water	a	b	c	d	e	f	g	h	i	¬a	¬b	¬c	¬d	¬e	¬f	¬g	¬h	¬i
Leech	×	×					×					×	×	×	×		×	×
Bream	×	×					×	×				×	×	×	×			×
Frog	×	×	×				×	×					×	×	×			×
Dog	×		×				×	×	×		×		×	×	×			
Spike–weed	×	×		×		×						×		×		×	×	×
Reed	×	×	×	×		×								×		×	×	×
Bean	×		×	×	×						×				×	×	×	×
Maize	×		×	×		×					×			×		×	×	×

Fig. 5. Living Beings and Water Context with Dichotomic Attributes

the object $d := (1, 0, 1, 0, 0, 0, 1, 1, 1)$, which happens to be the dog. The function $\vec{u}_d$ can be replaced by the Hamming distance

$$h_d(x) := \sum_{|x_i| \neq |d_i|} 1$$

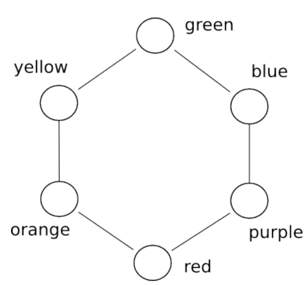
as we have $\vec{u}_d(x) \leq \vec{u}_d(y) \Leftrightarrow h_d(x) \leq h_d(y)$. The context graph shows the Hamming distance to the dog object next to each of the nodes. We can verify that the dog vertex can be reached from every other vertex via a decreasing path. As a concluding remark we want to note that the edge-minimal context graph on the right side is unique. More generally, the following proposition can be shown:

Proposition 1. *Let $\mathbb{M} = (G, \mathbb{S}_1, \ldots, \mathbb{S}_n)$ be a precontext with scales such that all $\mathbb{S}_i$ satisfy (P2). Then there is a unique edge-minimal context graph of $\mathbb{M}$, and the edges of this graph are given by*

$$x \sim y \Leftrightarrow \forall z \in G : \vec{u}_x(y) \leq \vec{u}_x(z) \ .$$

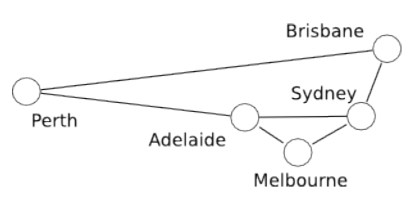
5 Scale Graphs

In this section we briefly present an idea how knowledge about domain of attribute values, or an individual perspective on such a domain, can be fed into a context graph. The idea is to first draw a graph which represents similarity among attribute values and then define a scale which has this graph as its context graph. We have no general method for this and just provide two examples, one defining a graph for an attribute "color" and one defining a graph for an attribute "location". The graphs are shown in Figs. 6 and 7. The extents have been chosen in a way that (P2) is satisfied, i.e. the extent generated by two objects is the set of all values "between" these objects. The extents can be read off from the scales which are shown next to the graphs. It can also be checked that the graphs are indeed context graphs of the given scales and that all edges are necessary. This means that if these scales are used for a precontext with scales $\mathbb{M}$, the structure of the scale graphs is reflected in any context graph K of $\mathbb{M}$. This approach allows to model the neighborhoods of a vertex in K. It would also be interesting to investigate the idea of embedding a context graph into a product of scale graphs, comparable to the subdirect product of lattices.



Colors	r-o-y	o-y-g	y-g-b	g-b-p	b-p-r	p-r-o
red	×				×	×
orange	×	×				×
yellow	×	×	×			
green		×	×	×		
blue			×	×	×	
purple				×	×	×

Fig. 6. Scale for a color wheel



Cities	PAM	PAS	PB	AM	ASB	MS	MSB
Perth	×	×	×				
Adelaide	×	×		×	×		
Melbourne	×			×		×	×
Sydney		×			×	×	×
Brisbane			×		×		×

Fig. 7. Cities in Australia

6 Conclusion

In this article, we have transferred the definition of context graphs into the more general setting where objects are described by many-valued attributes, using descriptions instead of attributes. Descriptions can be interpreted as queries of a user, although the queries could actually be formulated in some other language that the user is more comfortable with. In Sect. 4, we have investigated to what degree the structure of a context graph supports finding objects by their description. If the result set for a given query is nonempty, it may be preferrable to look up matching objects in a database and to use the graph structure for examining similar objects or result sets only. However, if the result set for a query is empty, it has been shown that best matches can be found by navigating through the graph. This may be an interesting starting point for interactive querying. Some search in the graph is unavoidable unless a decreasing path leads to a best match for a given query, which can only be guaranteed if a condition is met which comes at the price of further edges in the graph. Generally, navigation in the graph makes only sense if the number of edges is not too large. While a context graph can always be constructed in polynomial time [1], the capacity of the graph seems to be the limitation that applies to context graphs. We believe that the results presented in this paper provide an interesting point of reference but an evaluation of the theory within some real application scenario will be necessary.

References

1. Kötters, J., Schmidt, H., Squire, D.: Context graphs - representing formal concepts by connected subgraphs. In Ferré, S., Rudolph, S., eds.: Proceedings of ICFCA 2009. Volume 5548 of Lecture Notes in Artificial Intelligence., Berlin–Heidelberg, Springer (2009) 178–193
2. Gugisch, R.: Many-valued context analysis using descriptions. In Delugach, H.S., Stumme, G., eds.: Proceedings of ICCS 2001. Volume 2120 of Lecture Notes in Computer Science., Berlin–Heidelberg, Springer (2001) 157–168
3. Ganter, B., Wille, R.: Formal concept analysis: mathematical foundations. Springer-Verlag, Berlin (1999)
4. Godin, R., Missaoui, R., April, A.: Experimental comparison of navigation in a Galois lattice with conventional information retrieval methods. International Journal of Man-machine Studies **38** (1993) 747–767
5. Priss, U.: Lattice-based information retrieval. Knowledge Organization **27**(3) (2000) 132–142
6. Ferré, S.: CAMELIS: Organizing and browsing a personal photo collection with a logical information system. In Diatta, J., Eklund, P., Liquière, M., eds.: Proceedings of CLA 2007. Volume 331 of CEUR Workshop Proceedings. (2007) 112–123
7. Ducrou, J., Eklund, P.W.: SearchSleuth: The conceptual neighborhood of an web query. In Diatta, J., Eklund, P., Liquière, M., eds.: Proceedings of CLA 2007. Volume 331 of CEUR Workshop Proceedings. (2007)
8. Prediger, S.: Logical scaling in formal concept analysis. In Lukose, D., Delugach, H., Keeler, M., Searle, L., Sowa, J.F., eds.: Proceedings of ICCS 1997. Volume 1257 of Lecture Notes in Artificial Intelligence., Berlin–Heidelberg, Springer (1997) 332–341