

Many-valued Context Analysis using Descriptions

Ralf Gugisch

University of Bayreuth, Department of Mathematics,
D-95440 Bayreuth, Germany
`ralf.gugisch@uni-bayreuth.de`

Abstract. We propose an approach to many-valued contexts using formal descriptions instead of scaling. The underlying idea is the philosophical definition of a concept as a set of objects together with the *most precise description*.

We introduce a *formal description* as a mapping from the set of attributes to the power set of the values (which is extended appropriately to empty cells), assigning to each attribute the set of allowed values. Descriptions are naturally ordered by preciseness. Using this, we can introduce *extent* and *intent* according to the philosophical idea, and thus we define *concepts*. We present a way to restrict the amount of concepts for a many-valued context by *preselecting* some descriptions of interest. Furthermore, we introduce *implications* on descriptions, allowing to investigate relationships between attributes.

Within this approach, we reformulate the known theory under a different point of view. It certainly does not provide a better analysis than scaling, but it allows to avoid the generation of a huge one-valued context.

1 Many-valued Contexts

Recall the definition of many-valued contexts:

Definition 1. A many-valued context $\mathbb{K} = (G, M, W, I)$ is a set of objects G , a set of attributes M , a set of possible values W , and a ternary relation $I \subseteq G \times M \times W$, with

$$(g, m, w) \in I, (g, m, v) \in I \implies w = v .$$

$(g, m, w) \in I$ indicates, that object g has the attribute m with value w . In this case, we also write $m(g) = w$, regarding the attribute m as a partial function from G to W .

We can consider in particular each data base as a many-valued context, thus formal concept analysis appears as a tool of knowledge discovery and data-mining. Within this paper, we will consider the following small example (taken from [7]), representing some facts from algebra:

H. Delugach and G. Stumme (Eds.): ICCS 2001, LNAI 2120, pp. 157-168, 2001.
© Springer-Verlag Berlin Heidelberg 2001

Example 1. Let $\mathbb{K} = (G, M, W, I)$ be the many-valued context with the set of real number objects $G = \{2, \sqrt{2}, \sqrt[3]{2}, \pi\}$ together with the set of attributes $M = \{i, a, t\}$, where “i” means *irrational*, “a” *algebraic* and “t” *transcendental*. The set of values is $W = \{\times\} \cup \mathbb{N}$, and the incidence relation is specified by the table below.

	i	a	t
2		1	
$\sqrt{2}$	×	2	
$\sqrt[3]{2}$	×	3	
π	×		×

Irrational and transcendental are one-valued attributes, i.e. an “×” within the table indicates, that the given number has the corresponding property. But the attribute algebraic is many-valued: $n \in \mathbb{N}$ indicates, that this number is algebraic of degree n , i.e. it is a root of a polynomial over \mathbb{Q} of degree n , n being minimal with this property. If the given number is not algebraic, then we have an empty cell in the row of this number and in the column of the attribute “a”.

Formal concept analysis ([4]) was inspired by the fact that we are thinking in terms of concepts consisting of extent and intent. The extent is a set of objects and the intent is the set of attributes, which all these objects do have (according to the “logic of Port-Royal”, see [1]). We want to keep this orientation also for many-valued contexts, but we want to avoid the transformation of many-valued attributes into one-valued attributes.

For this purpose we quote Arnauld and Nicole, who write at the end of chapter VI:

“I call the intent of the idea (=concept) the (set of) attributes, which the idea comprises and which one cannot remove without destroying the idea,...”

This was meant for one-valued attributes. The known strategy to handle a many-valued context is, to scale it into a one-valued context (*plain scaling*) and to analyse the latter one considering concepts via extent and intent.

We want to present here an alternative method. It certainly yields the same results as plain scaling, but it gives a different (philosophically founded) interpretation to the concept lattices. Thus strictly speaking, we should not consider the present work as a “new strategy” or a “generalization” of formal concept analysis, rather than as a new interpretation of an already known strategy.

2 Descriptions

Considering many-valued contexts, we have, of course, to generalize the definition of concepts, and we should like to replace it by the following rough description of what we intend to make more precise later on:

A concept is a set of objects, together with the most precise description of them.

To begin with, let us briefly describe, where the problems are. Consider, for example, the attribute “red color”. It is by definition one-valued. An object either has the attribute “red color”, or not.

But if you want to do knowledge discovery and data-mining within a data base, you most often need to analyse attributes having several values. Instead of saying, that an object has a red color, we can also say that the color of the object is red. In the first case, we consider the one-valued attribute “red color”. In the latter form, objects can have different values in the many-valued attribute “color”: red, green, blue, etc. For data analysis, such many-valued attributes are better applicable than one-valued attributes.

But what is a description? For example, we may describe a car in a way like this: “It is a blue Ford Fiesta, 60-horsepower”. For computer representation, we would reformulate this information: “color: blue; model: Ford Fiesta; horsepower: 60”. In this way, we could store objects of type car within a data base, using the many-valued attributes color, model and horsepower.

If we want to describe a set of objects (for example the cars, we want to take into consideration for our next buy), we could specify the set of allowed values for each attribute. If we prefer, for example, the models VW Golf, or perhaps again a Ford Fiesta, we assign to the attribute “model” the set {“VW Golf”, “Ford Fiesta”}. If we state that our next car should have at least as many horsepowers as the old one, we assign an open interval $[60, \infty)$ to the attribute “horsepower”. If we do not have any preferences on the color, we assign the whole set of possible colors to the attribute “color”. (Normally, we would not explicitly mention the last aspect, as it is not really a restriction on the set of admissible cars.)

From the mathematical point of view, we assigned to each attribute the set of values, which are allowed for the objects we want to describe. Thus, in mathematical terms, we can define a *description* as a mapping from the set of attributes to the power set of values. (This is not the only possibility for describing objects, but the most simple one. If we need to consider relationships between attributes, we need a more powerful description, using a suitable logic. Work in this direction has been done in logical concept analysis, see [2], [3].)

With this idea of a description, we can specify, in which case an object g fulfills a description d , or equivalently, d holds for g , iff the value of g in each attribute m is contained in $d(m)$.

We also know a natural order on the set of descriptions: *preciseness*. A description is more precise (greater), iff it applies to less objects, i.e it restricts the values an object can have in some attribute.

Considering Empty Cells

The definition of descriptions so far would suffice, if each object would have in every attribute some value. Unfortunately, data bases most often have missing values. In the theory of formal concept analysis, they are called *empty cells*. In formal terms, many-valued contexts may contain pairs (g, m) with

$$(g, m, w) \notin I, \forall w \in W .$$

Such empty cells can have different interpretations: It could be, for example, that none of the values applies for the given object, or that the value is just not known.

We could try to handle empty cells just like an additional value, equivalent to the other ones. But this leads to unsatisfactory results. (In short, the effect is comparable to the consideration of the dichotomic situation of one-valued contexts, which is not wanted in every case. It would result in too many concepts. See [5] for more information.)

Thus, we need to consider empty cells in a different way: If an object g has no entry at attribute m (i.e. (g, m) is an empty cell), then a description should only hold for g if it has no restrictions on the values at attribute m . Especially, we need to distinguish between descriptions allowing all possible values at an attribute m , and descriptions additionally allowing, that there is no value given at all (i.e. allowing empty cells). The latter one is more general.

In order to handle this situation mathematically, we extend the power set of values, $\mathcal{P}(W)$, by adding a special maximal element ∞ :

$$\dot{\mathcal{P}}(W) := \mathcal{P}(W) \dot{\cup} \{\infty\} .$$

We extend the order on $\mathcal{P}(W)$ given by inclusion to $\dot{\mathcal{P}}(W)$ by defining

$$\begin{aligned} A \leq B, \forall A, B \in \mathcal{P}(W) \text{ with } A \subseteq B, \text{ and} \\ A < \infty, \forall A \in \mathcal{P}(W) . \end{aligned}$$

Furthermore, we regard each attribute $m \in M$ as a mapping from the

$$m: \mathcal{P}(G) \rightarrow \dot{\mathcal{P}}(W), m(A) := \begin{cases} \{m(g) | g \in A\} & \text{if } \nexists g \in A : (g, m) \text{ is empty cell,} \\ \infty & \text{otherwise.} \end{cases}$$

The notation $m(A)$ is motivated by the fact, that one often regards an attribute m as a (partial) mapping from G to W . Then, $m(A)$ denotes the image of m on the subset $A \subseteq G$. We extended this in order to handle empty cells. Furthermore, we now identify $m(g)$ with $m(\{g\})$. This is possible, as for nonempty cells (g, m) , we can identify the element $w := m(g)$ of W with the singleton $\{w\} \in \dot{\mathcal{P}}(W)$. For empty cells (g, m) , we have therewith $m(g) = \infty$.

Finally, we can give a mathematical definition for descriptions:

Definition 2. Let $\mathbb{K} = (G, M, W, I)$ be a many-valued context. A formal description d in \mathbb{K} is a mapping $d: M \rightarrow \dot{\mathcal{P}}(W)$ from the set M of attributes to the extended power set $\dot{\mathcal{P}}(W)$ of values. D denotes the set of all descriptions in \mathbb{K} :

$$D := \dot{\mathcal{P}}(W)^M = \{d: M \rightarrow \dot{\mathcal{P}}(W)\} .$$

This set is ordered by preciseness:

$$d_1 \leq d_2 :\iff d_1(m) \geq d_2(m), \forall m \in M ,$$

(where $d_1(m) \geq d_2(m)$ is defined with respect to the order in $\dot{\mathcal{P}}(W)$: either $d_1(m) = \infty$, or $d_1(m) \neq \infty, d_2(m) \neq \infty$ and $d_1(m) \supseteq d_2(m)$).
 An object $g \in G$ fulfills a description d , iff

$$m(g) \leq d(m), \forall m \in M .$$

(Again \leq is meant with respect to the order in $\dot{\mathcal{P}}(W)$). A description $d \in D$ holds for a set of objects $A \subseteq G$, iff each object $g \in A$ fulfills d .

One can show, that the set D together with the order of preciseness forms a complete lattice. The supremum $\bigvee d_i$ of a family of descriptions $d_i, i \in I$ is given by the conjunction of the descriptions, and the infimum $\bigwedge d_i$ is given by their disjunction:

$$\begin{aligned} \bigvee d_i &:= m \mapsto \bigcap d_i(m) , \\ \bigwedge d_i &:= m \mapsto \bigcup d_i(m) . \end{aligned}$$

(The used order is dual to the order normally taken when considering conjunction and disjunction. This perhaps confusing definition is motivated by the fact, that we are now able to introduce concepts like in standard formal concept analysis, specifying a Galois connection.)

3 The Concept Lattice

Now having an idea of what a description is, we can introduce concepts as pairs of extent and intent. Therefore, we associate with each set $A \subseteq G$ of objects its *intent* $A' \in D$, the most precise description holding for each $g \in A$. Vice versa, we associate with each description $d \in D$ its *extent* $d' \subseteq G$, the set of objects fulfilling d :

$$\begin{aligned} A' &:= M \rightarrow \dot{\mathcal{P}}(W), m \mapsto m(A) , \\ d' &:= \{g \in G \mid m(g) \leq d(m), \forall m \in M\} . \end{aligned}$$

The two mappings $\varphi : A \mapsto A'$ and $\psi : d \mapsto d'$ form a Galois connection between the power set of G and the lattice D of all descriptions (see [5] for a proof of this and the following results). Thus, we have a similar situation, as with one-valued contexts, and we can introduce the concept lattice as usual.

Definition 3. Let $\mathbb{K} = (G, M, W, I)$ be a many-valued context, and let D be the set of descriptions in \mathbb{K} . A formal concept of the context \mathbb{K} is a pair (A, d) , with $A \subseteq G, d \in D, A' = d$ and $d' = A$. We call A the *extent* and d the *intent* of the concept (A, d) . $\mathcal{B}(\mathbb{K})$ denotes the set of all concepts of the context \mathbb{K} .

The set of concepts is ordered by extent (which is dual to the order given by intent). In the case of $(A_1, d_1) \leq (A_2, d_2)$, we call (A_1, d_1) a *subconcept* of (A_2, d_2) , and (A_2, d_2) a *superconcept* of (A_1, d_1) . Furthermore, $\mathcal{B}(\mathbb{K})$ together with this order forms a complete lattice:

Theorem 1. $\mathcal{B}(\mathbb{K})$ is a complete lattice. Infimum und supremum for a subset of concepts $\{(A_i, d_i) \mid i \in I\} \subseteq \mathcal{B}(\mathbb{K})$ (with an arbitrary set of indices I) is given by

$$\begin{aligned} \bigwedge (A_i, d_i) &= (\bigcap A_i, (\bigcap A_i)') = (\bigcap A_i, (\bigvee d_i)''), \\ \bigvee (A_i, d_i) &= ((\bigwedge d_i)', \bigwedge d_i) = ((\bigcup A_i)'', \bigwedge d_i). \end{aligned}$$

Labelling the Concept Lattice

A concept lattice usually is visualized using line diagrams, labelled in a way allowing to identify the context with the concept lattice. In the following, we develop a suitable labelling of concept lattices for many-valued contexts.

The labelling with objects can be done in the same way, as with one-valued contexts: We write each object below the smallest concept, containing that object. Thus, we receive the extent of a concept as the union of all objects standing below a concept less than or equal to the given one.

In order to be able to label the concept lattice with descriptions in a clear way, we need to introduce a special subset of descriptions, let us call them *singular descriptions*. This set shall be large enough to represent each (other) description as a conjunction of singular ones. Thus, we call a description *singular*, iff

1. it has no restrictions on any attribute except for one attribute $m \in M$, and
2. for this attribute m , the only restriction is either, that it must have some value (i.e. it disallows empty cells), or that it must have a value unequal to $w \in W$ (also forbidding empty cells).

We give to these singular descriptions intuitive names: We call a description having no restrictions except that one attribute $m \in M$ must not be an empty cell, just like the attribute: m . One has to be a little bit careful with this nomenclature, as now m can stand for an attribute as well as for a description. But it should always be clear from the context, what was meant. Furthermore, if a description has no restrictions on attributes $\neq m$, but that m must neither be an empty cell, nor equal to the value w , we call that restriction $m_{\neq w}$.

$$\begin{aligned} m : M \rightarrow \dot{\mathcal{P}}(W), \quad m(n) &:= \begin{cases} W & \text{if } n = m, \\ \infty & \text{else.} \end{cases} \\ m_{\neq w} : M \rightarrow \dot{\mathcal{P}}(W), \quad m_{\neq w}(n) &:= \begin{cases} W \setminus \{w\} & \text{if } n = m, \\ \infty & \text{else.} \end{cases} \end{aligned}$$

The set of all singular descriptions $\{m \mid m \in M\} \cup \{m_{\neq w} \mid m \in M, w \in W\}$ forms a \vee -dense subset of D , that means, we can represent each description as conjunction of singular descriptions.

Using this, we can label the concept graph with descriptions, too: We write the singular descriptions above the greatest concept they appear in. Then we can get the intent of a concept as conjunction of all singular descriptions standing above a concept greater or equal to the given one.

The concept lattice of the many-valued context for real numbers from Example 1 can be seen in Fig. 1.

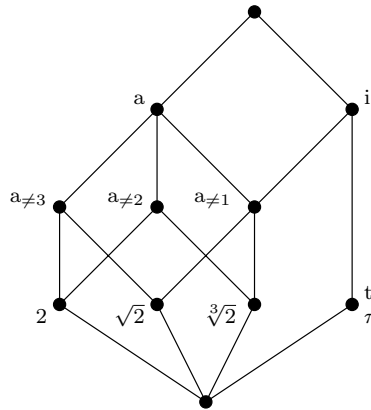


Fig. 1. The concept lattice of the context for real numbers

4 Preselection of Descriptions

An analysis of the concept lattice of the example above shows, that it contains some concepts, we are hardly interested in. For example, the context labeled by the description $a_{\neq 2}$ contains all algebraic numbers of degree 1 or of degree ≥ 3 . Normally, we are only interested in concepts whose intent restricts the algebraic degree to some interval.

This effect gets worse, if we consider bigger many-valued contexts. Various unreasonable concepts would arise and make the concept lattice unnecessary complex.

The problem is, that our set D of descriptions in \mathbb{K} can contain descriptions assigning attributes to abnormal sets of allowed values without any reasonable interpretation. Having such undesired descriptions, we may also get concepts using them.

But it is known from lattice theory, that we can restrict the derivation operators to a \vee -subsemilattice $\bar{D} \subseteq D$ of the set of descriptions simply by defining the intent of a set $A \subseteq G$ of objects as the most precise description in \bar{D} (instead of the most precise one in D), whereas the definition of the extent remains the same. This way, we obtain another set of concepts, having only intents from \bar{D} . These concepts form a \wedge -subsemilattice of the original concept lattice.

Thus we just need to specify an arbitrary subset $\tilde{D} \subseteq D$, say, a *preselection* of descriptions of interest. Then we define \bar{D} as the closure of \tilde{D} under conjunction:

$$\bar{D} := \{ \bigvee B \mid B \subseteq \tilde{D} \} .$$

This way, we can influence the amount of descriptions taken into consideration as intent for concepts. We can exclude a lot of unreasonable concepts, and we can do problem oriented analysis into specific aspects of a many-valued context. We call the resulting subsemilattice of concepts a *scaled concept lattice* with respect to the preselection \tilde{D} .

We label a scaled concept lattice with the preselected descriptions $d \in \tilde{D}$. As these are \vee -dense in \bar{D} , and as each intent in the scaled concept lattice is contained in \bar{D} , this labelling is sufficient. We still receive the intent of each concept as the conjunction of preselected descriptions standing above concepts greater than or equal to the considered one.

In order to have intuitive labellings in the scaled concept lattice, we define some standard descriptions usable in preselections. For most purposes, it will suffice to consider within preselections only description restricting one attribute $m \in M$, i.e. assigning ∞ to all other attributes.

The easiest kind of such descriptions are *nominal descriptions*, allowing for m only one value $w \in W$:

$$m_{=w} : M \rightarrow \dot{\mathcal{P}}(W), m_{=w}(n) := \begin{cases} \{w\} & \text{if } n = m, \\ \infty & \text{else.} \end{cases}$$

Furthermore, if an attribute has ordinal values, we can consider *ordinal descriptions*, allowing open intervals for the attribute m :

$$m_{\leq w} : M \rightarrow \dot{\mathcal{P}}(W), m_{\leq w}(n) := \begin{cases} (-\infty, w] & \text{if } n = m, \\ \infty & \text{else.} \end{cases}$$

$$m_{\geq w} : M \rightarrow \dot{\mathcal{P}}(W), m_{\geq w}(n) := \begin{cases} [w, \infty) & \text{if } n = m, \\ \infty & \text{else.} \end{cases}$$

In the same way, one could define further kinds of standard descriptions if necessary.

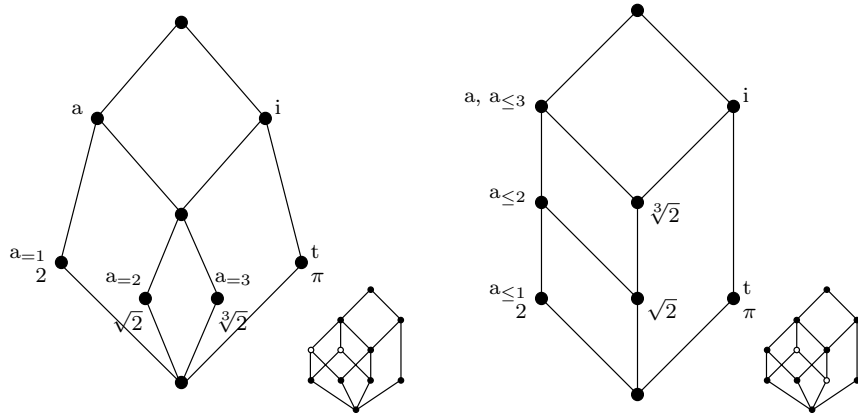


Fig. 2. Two scaled concept lattices of the context for real numbers. The embedding into the total concept lattice is indicated by the small pictures

Reconsider Example 1: Figure 2 shows two possibilities of scaled concept lattices. In the first one, we allowed only nominal descriptions for the attribute

algebraic, i.e. we used the preselection $\tilde{D}_1 = \{i, a, t, a_{=1}, a_{=2}, a_{=3}\}$. The second concept lattice results, if we allow only ordinal descriptions of the type $a_{\leq w}$, i.e. we used $\tilde{D}_2 := \{i, a, t, a_{\leq 1}, a_{\leq 2}, a_{\leq 3}\}$.

Both scaled concept lattices can be embedded into the original one. This is indicated in Fig. 2 by the small pictures beside the concept lattices. As one can recognize, this embedding is infimum-preserving: the infimum of two full nodes is full, again.

Both scaled concept lattices contain less concepts as the original one, and so they are less complex. But they emphasize different aspects of the context.

5 Implications

In order to analyse relationships between attributes of a formal context, we introduce implications between descriptions. Given a many-valued context $\mathbb{K} = (G, M, W, I)$, an implication $a \rightarrow b$ between two descriptions $a, b \in D$ holds in \mathbb{K} , iff each object $g \in G$, which fulfills description a , also fulfills description b . More formally, we define:

Definition 4. Let $\mathbb{K} = (G, M, W, I)$ be a many-valued context, and let $D = \mathcal{P}(W)^M$ be the set of all descriptions within \mathbb{K} . Furthermore, let $a, b \in D$ be descriptions.

A description $d \in D$ respects the implication $a \rightarrow b$, if $a \not\leq d$ or $b \leq d$. d respects a set \mathcal{L} of implications if d respects every single implication in \mathcal{L} . $a \rightarrow b$ holds in a set $\{d_1, d_2, \dots\}$ of descriptions if each of the descriptions d_i respects the implication $a \rightarrow b$. $a \rightarrow b$ holds in the context \mathbb{K} if it holds in the system of object intents. In this case, we also say, that $a \rightarrow b$ is an implication of the context \mathbb{K} or, equivalently, that within the context \mathbb{K} , a is a premise of b .

We defined an equivalent structure to implications between attributes in one-valued contexts. The known algorithms and further definitions like the stem base can be translated in a straightforward way to many-valued contexts and implications between descriptions, as shown in [5].

We can represent descriptions as conjunctions of singular or preselected ones. Thus we specify a description d just by listing all singular/preselected descriptions smaller than d .

As an implication $a \rightarrow b$ is always equivalent to $a \rightarrow a \vee b$, we don't need to repeat on the right hand side singular/preselected descriptions already standing on the left hand side of an implication. Thus, we can represent an implication $a \rightarrow b$ in a reduced way, listing on the left hand side all singular/preselected descriptions smaller than a , and on the right hand side all additional ones smaller than b .

Example 2. Reconsider Example 1 for real numbers. The stem base is:

$$\begin{aligned} t &\rightarrow i \\ a_{\neq 1} &\rightarrow i \\ i, a &\rightarrow a_{\neq 1} \\ i, a_{\neq 1}, t &\rightarrow M \end{aligned}$$

As already known, we can consider specific aspects of the concept lattice using a preselection. Then, we also get a different stem base. Below we show the stem bases resulting from the two preselections using nominal or ordinal descriptions for the attribute algebraic, as already considered above (Example 4):

$$\begin{array}{ll}
 t \rightarrow i & t \rightarrow i \\
 a_{=3} \rightarrow i & a \rightarrow a_{\leq 3} \\
 a_{=2} \rightarrow i & i, a_{\leq 3}, t \rightarrow M \\
 i, a, t \rightarrow M & i, a_{\leq 1} \rightarrow M \\
 i, a_{=1} \rightarrow M &
 \end{array}$$

Using implications on descriptions, we can apply attribute exploration directly on many-valued contexts.

6 Relationship to One-valued Contexts

It is well known, that one can generate one-valued contexts out of many-valued ones using scales. What is the relationship between these scaled contexts and the approach on many-valued contexts given in this paper?

First, we can specify for each many-valued context $\mathbb{K} = (G, M, W, I)$ a one-valued context $\mathbb{K}' = (G', M', I')$ having an isomorphic concept lattice: Just take the set of objects, $G' := G$, as set of attributes M' take the singular descriptions, and let (g, d) be in I' iff g fulfills d . If you want to use a preselection \tilde{D} , you get a one-valued context with isomorphic concept lattice by taking the preselected descriptions as M' instead of the singular ones.

A closer look at these one-valued contexts associated with preselections shows, that one can interpret each plain scaling as a preselection. As a consequence, we can interpret plain scaling as a subcontext of the many-valued context, having as concept lattice an \wedge -subsemilattice of the many-valued concept lattice.

On the other hand, preselections can be understood as scaling, thus it provides no better analysis results.

This approach can be seen as a natural extension of one-valued contexts from another point of view, too: If we interpret a one-valued context $\mathbb{K} = (G, M, I)$ as a many-valued one with exactly one value " \times ": $\mathbb{K}' := (G, M, \{\times\}, I')$ with

$$(g, m, \times) \in I' \Leftrightarrow (g, m) \in I,$$

then the concept lattices of \mathbb{K} and of \mathbb{K}' are isomorphic.

This can be seen as follows: Considering singular descriptions in \mathbb{K}' , the only nontrivial ones are of the form $m, m \in M$, as the descriptions of the form $m_{\neq \times}$ do not hold for any object $g \in G$. (If an object g has a value in attribute m , i.e. (g, m) is no empty cell, it must have the only value \times .) Thus, as we have seen above, \mathbb{K}' is isomorphic to the one-valued context $\mathbb{K}'' = (G, \{m, m \in M\}, I'')$ with

$$(g, m) \in I'' \Leftrightarrow g \text{ fulfills } m \Leftrightarrow (g, m, \times) \in I' \Leftrightarrow (g, m) \in I.$$

Thus I'' is equivalent to I , and \mathbb{K}'' equivalent to \mathbb{K} .

Another point worth to consider is the comparison between implications on descriptions and implications on one-valued attributes, i.e. scaling a one valued context and considering implications within the scaled context. The second strategy may lead to trivial implications representing relationships between different states of one many-valued attribute.

Within our example for real numbers, the stem base of a scaled context using the ordinal scale for attribute algebraic needs to express implications like $a_{\leq 2} \rightarrow a_{\leq 3}$. Using descriptions, we do not need to express such implications, as we know the order on descriptions, which means that we know some implicit background information.

There exists some approach ([8]) considering background information for implications on attributes. Using this, one should obtain similar stem bases with both strategies. But it remains to do some research on this aspect.

7 Conclusion

We have introduced a theory of many-valued concept analysis using descriptions. This provides a different way to analyse concepts within many-valued contexts, besides the possibility of scaling them and analysing the resulting one-valued context. Thanks to the possibility of preselecting specific descriptions, the new strategy remains as flexible as scaling, and it turns out, that it is possible to get equivalent results with both strategies.

One advantage of using descriptions is that we avoid generating huge one-valued contexts. Furthermore, we are able to consider implications on descriptions, and therefore may apply strategies like attribute exploration directly on many-valued contexts.

Acknowledgements. Special thanks are due to A. Kerber for his continuous support, and to the referees for very helpful hints.

References

1. Arnauld, A. Nicole, P.: *La Logique ou L'Art de penser, contenant, outre les Règles communes, plusieurs observations nouvelles, propres à former le jugement*. Sixième Édition revue et de nouveau augmentée. A Amsterdam, Chez Abraham Wolfgang (1685). German translation by Axelos, C., 2. edition, Wiss. Buchges., Darmstadt (1994)
2. Chaudron, L., Maille, N.: *Generalized Formal Concept Analysis*. In: Ganter, B., Mineau, G.W., editors, *Conceptual structures: Logical, Linguistic, and Computational Issues*. Lecture Notes in Artificial Intelligence 1867. Springer, Berlin-Heidelberg-New York (2000) 357–370
3. Ferre, S., Ridoux, O.: *A Logical Generalization of Formal Concept Analysis*. In: Ganter, B., Mineau, G.W., editors, *Conceptual structures: Logical, Linguistic, and Computational Issues*. Lecture Notes in Artificial Intelligence 1867. Springer, Berlin-Heidelberg-New York (2000) 371–384

4. Ganter, B., Wille, R.: *Formal Concept Analysis – Mathematical Foundations*. Springer Verlag, Berlin (1999)
5. Gugisch, R. *Verallgemeinerung der formalen Begriffsanalyse und Anwendung auf mehrwertige Kontexte*. Diplomarbeit, Uni Bayreuth (1997)
<http://www.mathe2.uni-bayreuth.de/ralfg/papers/diplom.ps.gz>
6. Gugisch, R. *Lattice Contexts – a Generalization in Formal Concept Analysis*. hand-out to ICCS 2000, Darmstadt (2000)
<http://www.mathe2.uni-bayreuth.de/ralfg/papers/iccs2000.ps.gz>
7. Kerber, A., Lex, W.: *Kontexte und ihre Begriffe*. (1997)
<http://www.mathe2.uni-bayreuth.de/kerber/begriffe.ps>
8. Stumme, G.: *Attribute exploration with background implications and exceptions*. In: Bock, H.-H., Polasek W., editors, *Data analysis and information systems*. Springer, Berlin-Heidelberg-New York (1996) 457-469