

# Navigation Through Query Result Using Concept Order

Václav Snášel, Tomáš Skopal, Daniela Ďuráková

Department of Computer Science, VŠB–Technical University Ostrava, Czech Republic  
`vaclav.snasel@vsb.cz`, `tomas.skopal@vsb.cz`, `daniela.durakova@vsb.cz`

**Abstract.** Query in Information Retrieval produces some amount of relevant results. Consecutively, there is a need for some qualitative classification of these particular results in such way the user is able to understand. In this article we introduce a new formal method of navigation through query result. This navigation method is based on an original idea of concept order structure, which exploits the concept lattices theory and the fuzzy set theory. So far, user must provide a subjective factor – attribute scaling. Our method helps to uncover significant concepts without the need of user scaling.

**Keywords:** concept lattice, navigation, fuzzy set, query, alpha cut

## 1 Introduction

Strict boolean query interpretation does expect as the result a set of such objects, each particular property of which matches given condition. From the query-semantic point of view, all these objects in query result are equally important.

However, in Information Retrieval (IR), we must often consider notions like *query uncertainty* and *response relevancy*. Query uncertainty is the typical attribute of IR. In IR we are not able to predetermine exactly which particular property of object in result is to match any predefined condition. We can only specify a measure of relevancy the object must satisfy. Thus, strict queries are not quite suitable for IR purposes.

Let us imagine as an example a task to find the most suitable ski centre for our holidays. We live in Czech Republic and prefer alpine ski centers that are near, are situated on the highest elevations and are selling inexpensive ski-pass. Had we express a strict query on those ski centers (objects) each property of which must satisfy the best possible value, we wouldn't probably receive any result. However, such answer is insufficient for us. Thus, we formulate a query very loosely – "alpine ski centre". This query now produces several (actually much more, but for simplicity nine) ski centers (see Table 1). On the basis of this result we want to obtain the "best compromise". A way to find this compromise can be called "navigation through query result".

**Table 1.** Query result of the ski centers

Ski Centre	Abbreviation	<i>distance</i> (km)	<i>ski-pass</i> price (CZK)	<i>elevation</i> (m)
Mayrhofen	Ma	476	5276	3250
Sölden	So	576	4866	3260
Kitzbühel	Ki	455	4741	2000
Flattach	Fl	490	4411	3160
Söll	Sl	453	3664	1835
Zell am See	Ze	475	4632	3029
Radstadt	Ra	450	4625	2130
Gosau	Go	390	3774	1600
Rohrmoos	Ro	426	4565	2700

### 1.1 Existing Approaches

During the time, there were various methods of navigation evolved and some of them exploit properties of concept lattices. Concept lattice is an algebraic structure which makes possible to order a set of objects with attributes.

Concept lattice theory is based on classical set theory and therefore a *crisp* set of attributes is assigned to each object. Object either *has* or *has not* a particular attribute. For example, object "dog" has attribute "fur" but doesn't have attribute "flippers". However, in real problems there exist attributes of *many-valued* nature, e.g. object "car" could own numerical attribute "fuel consumption". This restriction was attempted to be removed with the introduction of *attribute scaling*, e.g. attribute "fuel consumption" might be distributed over three attributes "low f.c.", "middle f.c." and "high f.c.". Afterwards, object "car" shall satisfy some of the three new attributes. Furthermore, conventional *fuzzyfication* of attributes solves this problem in a similar manner (e.g. [1]).

### 1.2 Briefly Our Solution

Unfortunately, there is an unpleasant consequence of the conventional attribute scaling or fuzzyfication that makes these methods worse applicable. It is because the scaling phase "swells" the set of attributes too much and the resultant lattice is very large and confusing structure.

Our alternative is based on concept lattices and fuzzyfication as well, but in a different way. Instead of scaling the set of attributes, we normalize their values into interval  $[0,1]$  – in other words we create appropriate fuzzy set of objects for each attribute. From the obtained fuzzy sets we construct so called *fuzzy context* and define  $\alpha$ -cuts. For each  $\alpha$ -cut we derive its crisp context and its concept lattice. We unify "somehow" all the produced concept lattices (their concepts respectively) and obtain an ordered structure of concepts as a result. This structure is then more compact and more readable than classical concept lattice. Furthermore, significant concepts in this structure are scale independent. Simultaneously, there is minimal loss of important characteristics. Additional theoretical background of this method can be found in [8].

## 2 Basics of Concept Lattices Theory

The idea of formalization of terms *context* and *concept* by means of lattices theory is not quite new. Formerly isolated experiments at application so called Galois lattices already existed, especially in the area of Information retrieval, but systematically built up theory did not arise until the work of R.Wille and his group [10]. For the formalization of term concept and context, the theory of ordered sets and the theory of lattices are applied. The terms from these classic disciplines can be found in [2].

**Definition 1.** A *context* is a triple  $(G, M, I)$ , where  $G$  and  $M$  are sets and  $I \subseteq G \times M$ . The elements of  $G$  are called *objects* and the elements of  $M$  are called *attributes*. We write  $gIm$  or  $(g, m) \in I$  and say "the object  $g$  has the attribute  $m$ ". The relation  $I$  is called *incidence relation* of context. Contexts of a smaller extent are simply representable by a table.

**Definition 2.** For  $A \subseteq G$  and  $B \subseteq M$ , define

$$\begin{aligned} A^\uparrow &= \{m \in M \mid gIm \ \forall g \in A\} \\ B^\downarrow &= \{g \in G \mid gIm \ \forall m \in B\} \end{aligned}$$

where  $A^\uparrow$  is the set of attributes common to all objects in  $A$  and  $B^\downarrow$  is the set of objects possessing the attributes in  $B$ .

Two operators were defined

$$\begin{aligned} \uparrow &: P(G) \rightarrow P(M) \\ \downarrow &: P(M) \rightarrow P(G) \end{aligned}$$

where  $P(G)$  and  $P(M)$  are the set of all context of the sets  $G$  and  $M$  respectively.  $B^{\downarrow\uparrow}$  indicates a set  $(B^\downarrow)^\uparrow$ , that means double application of operator from the previous definition.

**Definition 3.** Let  $(G, M, I)$  be a context. For couple  $(A, B)$  where  $A \subseteq G$ ,  $B \subseteq M$ ,  $A^\uparrow = B$  and  $B^\downarrow = A$  we say  $A$  is *extent* and the set  $B$  is *intent* of concept  $(A, B)$ . The set of all concepts is called  $\mathcal{K}(G, M, I)$ .

**Definition 4.** For concepts  $(A_1, B_1)$  and  $(A_2, B_2)$  in  $\mathcal{K}(G, M, I)$  we say  $(A_1, B_1)$  is a *subconcept* of  $(A_2, B_2)$  if  $A_1 \subseteq A_2$  (it is the same as  $B_2 \subseteq B_1$ ) and we write  $(A_1, B_1) \leq (A_2, B_2)$  and that  $(A_2, B_2)$  is a *superconcept* of  $(A_1, B_1)$ .

The relation  $\leq$  is the relation of order on the set of all concepts  $\mathcal{K}(G, M, I)$ . This relation is even lattice order on this set, that means that there exist supremum and infimum with regard to  $\leq$  for every two elements in  $\mathcal{K}(G, M, I)$ . The proof and further details can be found in [2].

**Proposition 1.** Let  $m$  be the object count and  $n$  the attribute count of a context. Then concept count  $c_{\mathcal{K}}$  of given context is limited by following statement

$$c_{\mathcal{K}} \leq 2^{\min(m, n)} \quad (1)$$

This inequality has important influence on further analysis as we will see in the next section.

### 2.1 Scaling

Traditional methods of scaling (described in [2]) are based on decomposition of many-valued attribute onto several single-valued attributes. Moreover, these new attributes must preserve some specific relations between each other.

- *Nominal scales* are suitable to scale attributes, the values of which mutually exclude each other.
- *Ordinal scales* are used to scale many-valued attributes, the values of which are ordered and where each value implies the weaker ones. This type is also suitable for fuzzy scaling because the higher membership values always imply the lower ones.
- *Interordinal*, *biordinal* and other existing scales are more complex and we don't consider them in this paper.

**Table 2.** Nominal scale

president	nationality		
	<i>czech</i>	<i>german</i>	<i>russian</i>
Havel	×		
Putin			×
Rau		×	

**Table 3.** Ordinal scale

thing	temperature		
	<i>warm</i>	<i>hot</i>	<i>burning</i>
tea	×		
stove	×	×	
lava	×	×	×

### 3 Applications Using Scaling

Let's now apply scaling on our ski centre problem. According to rules of ordinal scaling and the data in Table 1, we create crisp *scaled context*.

**Table 4.** Scaled context of ski centers

Ski centre	distance				s.p. price				elevation			
	$\vee_1$	$\vee_2$	$\vee_3$	$\vee_4$	$\vee_1$	$\vee_2$	$\vee_3$	$\vee_4$	$\wedge_1$	$\wedge_2$	$\wedge_3$	$\wedge_4$
	$d_1$	$d_2$	$d_3$	$d_4$	$s_1$	$s_2$	$s_3$	$s_4$	$e_1$	$e_2$	$e_3$	$e_4$
Ma	×	×							×	×	×	×
So									×	×	×	×
Ki	×	×	×		×							
Fl	×				×	×			×	×	×	
Sl	×	×	×		×	×	×		×			
Ze	×	×			×				×	×	×	
Ra	×	×	×		×				×			
Go	×	×	×	×	×	×	×	×				
Ro	×	×	×	×	×	×			×	×		

Concept lattice (see Figure 1) of the above context has 36 concepts. Each node in lattice (concept) represents a *navigation decision* and each edge in lattice (concept order) allows a *navigation movement*.

**Note:** The concepts in Figure 1 are labeled according to the *reduced labeling* making the lattice more readable (see [2] for more detail).

### Example of navigation:

In Figure 1, object concept *Ra* is greyed. This concept is our initial navigation decision and it means that we choose ski centres *Ra*, *Ro*, *Sl* that are very near (attributes  $d_1, d_2, d_3$ ), are quite expensive (attribute  $s_1$ ) and are quite low elevated (attribute  $e_1$ ). Three arrows pointing out from this concept represent possible navigation movements. If we choose the downward arrow we will obtain such concept as a navigation decision that has lost ski centre *Ra* but has gained attribute  $s_2$ . In other words, with this movement only two possible ski centres remain but they sell quite cheap ski-pass.

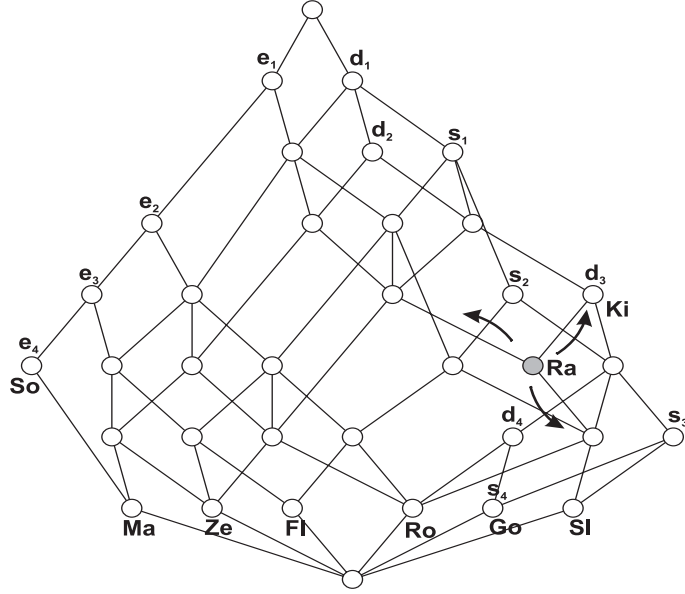


Fig. 1. Concept lattice for scaled context

### 3.1 Drawbacks

Using of attribute scaling brings following major drawbacks:

#### – large structure volume

From the proposition on the beginning of this section we can expect that the growing set of attributes implies (in the worst case) exponential growth of the resultant concept lattice. This claim is well-founded because the set of objects is supposed to be always greater then the set of attributes, which is obvious due to the nature of query result.

#### – dependency on scaling

Here are two aspects:

- At the beginning, user must provide a scale for each attribute. There is no general method how to determine ideal scaling.
- If we modify (even slightly) the set of attributes during the scale refinements, the resultant lattice will change without the preservation of the most significant concepts in the former structure.

### 3.2 Other Research

There were attempts to bring the fuzzy factor on the stage (see [1]). However, resultant concept lattice structures incorporate big number of concept nodes, similarly like by attribute scaling. This consequence stands such approaches on the attribute scaling level.

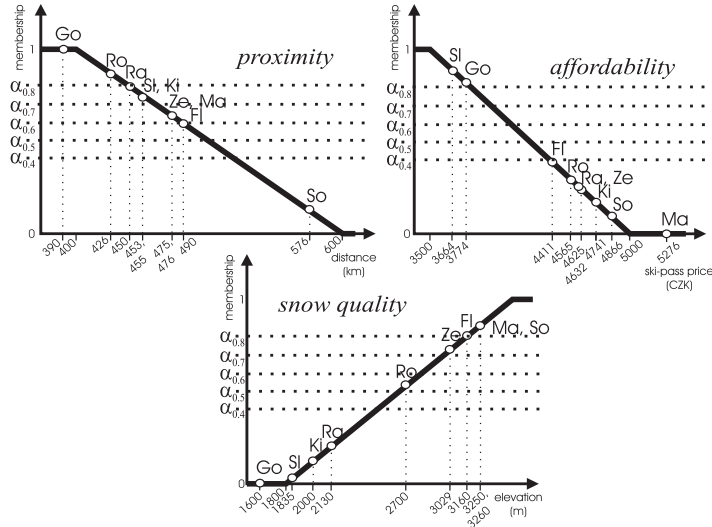
## 4 Our Idea

As we can see, the solution in the previous section is strongly tied with attribute scaling. Our motivation was to search such concepts that are scale independent and bring better support to the user. So far, the problem of "good" scaling was left upon the user's subjective meaning. This aspect was formerly objected to us (in [5]), thus we also tried to minimize the problem of *subjectivity*. The idea is to remove the scale factor that brings large lattice volume and dependency on attribute scales.

### 4.1 Context Normalization

First, we tried to formalize already the initial problem of object memberships to attributes. For each attribute we create a fuzzy set of objects. This set is constructed as follows:

1. Membership boundaries are chosen. By user or automatically.
2. Membership function is chosen. By user or automatically.



**Fig. 2.** Fuzzy sets *proximity* (*p*), *affordability* (*a*), *snow quality* (*q*) and  $\alpha$  – cuts

For automatic fuzzy set generation we can choose the smallest and the greatest (in sense of attribute measure) object of all attributes as the membership boundaries. However, for our ski centre example we have chosen "user defined" boundaries [400, 600] for distance, [3500, 5000] for ski-pass price and [3500, 1800] for elevation. Furthermore, linear membership function was chosen. See Figure 2.

Second, it is easy now to transform query result through fuzzy sets into *fuzzy context*. In this context, the attribute values are the fuzzy set membership values of the original attributes. See Table 5.

**Table 5.** Normalized fuzzy context for ski centers

Ski Centre	<i>proximity</i>	<i>affordability</i>	<i>snow quality</i>
Ma	0.61	0.00	0.85
So	0.12	0.09	0.86
Ki	0.73	0.17	0.12
Fl	0.55	0.39	0.80
Sl	0.74	0.89	0.02
Ze	0.62	0.24	0.72
Ra	0.75	0.25	0.19
Go	1.00	0.82	0.00
Ro	0.87	0.29	0.53

#### 4.2 Contexts of $\alpha$ -Cuts

On the fuzzy context (on defined fuzzy sets respectively) we create  $\alpha$ -cuts (see Figure 2). From the  $\alpha$  value and the fuzzy context we are able to derive a crisp context for each of the particular  $\alpha$ -cuts, i.e.  $\alpha$ -context. For our ski centre problem we derive contexts for  $\alpha$ -cuts e.g.  $\alpha_{0.4}$ ,  $\alpha_{0.5}$ ,  $\alpha_{0.6}$ ,  $\alpha_{0.7}$  and  $\alpha_{0.8}$ .

Context for  
 $\alpha_{0.4}$ -cut

obj	p	a	q
Ma	x		x
So			x
Ki	x		
Fl	x	x	
Sl	x	x	
Ze	x		x
Ra	x		
Go	x	x	
Ro	x		

Context for  
 $\alpha_{0.5}$ -cut

obj	p	a	q
Ma	x		x
So			x
Ki	x		
Fl	x	x	
Sl	x	x	
Ze	x		x
Ra	x		
Go	x	x	
Ro	x		x

Context for  
 $\alpha_{0.6}$ -cut

obj	p	a	q
Ma	x		x
So			x
Ki	x		
Fl			x
Sl	x	x	
Ze	x		x
Ra	x		
Go	x	x	
Ro	x		

Context for  
 $\alpha_{0.7}$ -cut

obj	p	a	q
Ma			x
So			x
Ki	x		
Fl			x
Sl	x	x	
Ze			x
Ra	x		
Go	x	x	
Ro	x		

Context for  
 $\alpha_{0.8}$ -cut

obj	p	a	q
Ma			x
So			x
Ki			
Fl			x
Sl		x	
Ze			
Ra			
Go	x	x	
Ro	x		

### 4.3 $\alpha$ -Concept Order

What can we imagine behind particular  $\alpha$ -contexts? Concept lattice of a particular  $\alpha$ -context with a high  $\alpha$  value contains important concepts, the objects of which highly satisfy concept's attributes. On the other side, concept lattice of  $\alpha$ -context with a low  $\alpha$  value contains also those concepts, the objects of which satisfy the concept's attributes only a little.

Combination of all that concepts and their consequential ordering will produce relatively small plastic structure, i.e.  $\alpha$ -concept order, which makes navigation easier.

**Definition 5.** ( $\alpha$ -concept)

Let  $\mathcal{K}_\alpha$  be union of all the concepts produced from all  $\alpha$ -contexts. Then all concepts  $c_i, \dots, c_k \in \mathcal{K}_\alpha$  such that  $G(c_i) = \dots = G(c_k)$  form  $\alpha$ -concept  $c_\alpha = (G(c_\alpha), M(c_\alpha))$ . Set of objects  $G(c_\alpha)$  is determined by the previous condition. Set of attributes  $M(c_\alpha)$  is constructed as a multiset  $M(c_i) \cup_m \dots \cup_m M(c_k)$  which means that to each attribute in the union is assigned a *weight* – i.e. occurrence count within sets  $M(c_i), \dots, M(c_k)$ . Weight of  $\alpha$ -concept is overall weight of all its attributes.

**Table 6.**  $\alpha$ -concepts of certain 6  $\alpha$ -contexts

$\alpha$ -concepts	Objects										Attributes		
	$Ma$	$So$	$Ki$	$Fl$	$Sl$	$Ze$	$Ra$	$Go$	$Ro$		p	a	q
0	x	x		x		x			x				2
1	x		x	x	x	x	x	x	x		2		
2	x			x		x			x		2		2
3				x	x			x			1	1	
4				x							1	1	1
5	x	x	x	x	x	x	x	x	x				
6					x			x			3	4	
7											6	6	6
8	x	x		x		x							2
9	x		x		x	x	x	x	x		1		
10	x					x					1		1
11			x		x		x	x	x		1		
12	x	x		x								1	
13								x	x			1	
14								x			3	1	
15					x							1	

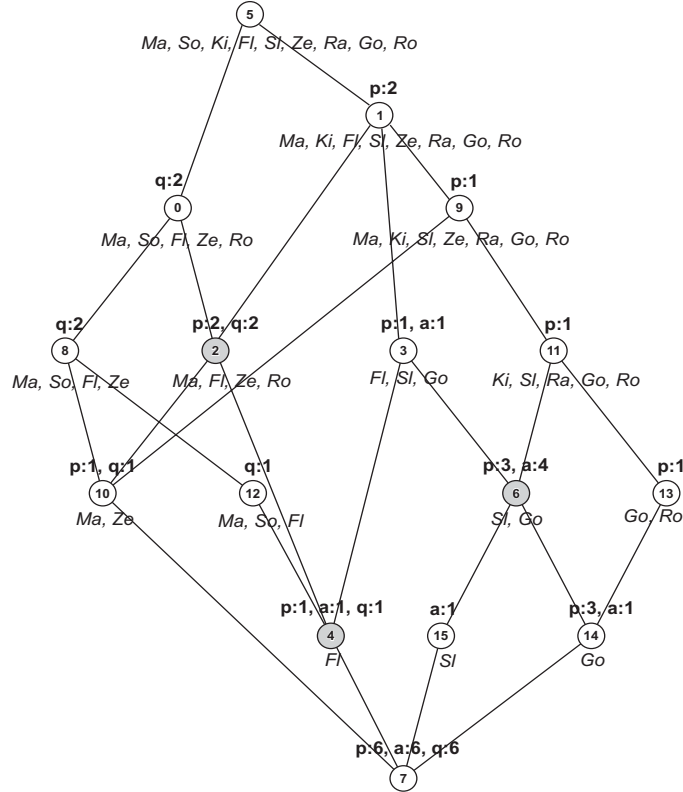
**Definition 6.** ( $\alpha$ -concept order)

All  $\alpha$ -concepts derived from  $\mathcal{K}_\alpha$  can be ordered according to  $G(c_\alpha)$  inclusion.



**Note:**

A remarkable property of highly weighed  $\alpha$ -concept is an analogy to the concept of crisp context. A higher weight of an  $\alpha$ -concept means that the concept appears simultaneously in more  $\alpha$ -contexts (in their lattices respectively). Thus,  $\alpha$ -concepts with highest weights could be considered as concepts of crisp contexts where the attributes have only "yes or no" values. From this point of view,  $\alpha$ -concept with high weight can be considered as *significant* or *important*.



**Fig. 3.**  $\alpha$ -Concept order for  $\alpha$ -concepts in Table 6.

**Example of navigation** in  $\alpha$ -concept order:

In Figure 3 we can see three grayed concept nodes which may be considered by the user as the significant ones. Concepts 2,6 have high weights while the concept 4 satisfies all the attributes (regardless their lower weights).

Further navigation decisions and movements relate to various user interpretations similarly like the navigation described in section 3.

#### 4.4 Selection of the $\alpha$ -Cuts

Significant  $\alpha$ -concepts have an interesting property. They appear simultaneously in many contexts of  $\alpha$ -cuts. This knowledge indicates that anyhow fine the di-

vision of  $\alpha$ -cuts is, the significant  $\alpha$ -concepts remain preserved. In other words,  $\alpha$ -concept order structure is independent on  $\alpha$ -cut selection.

Similar outcomes with  $\alpha$ -cuts has published M.Schneider who studied the influence of  $\alpha$ -cut selection on the values of fuzzy topological predicates (see [4]).

## 5 Conclusions

We have presented alternative method of navigation through query result which is based on well-known basis of concept lattices theory and fuzzy set theory.

We want to emphasize two main highlights of our solution. First, suggested apparatus offers to the user preferentially those concepts (decisions) which do not depend on certain attribute scaling. Second, navigation structure is compact and readable without the loss of important characteristics.

Additional details:

- Algebraic properties of the  $\alpha$ -concept structure can be described using  $\lambda$ -lattice (see [6], [7]).
- Described navigation method is completely implemented and is being tested.

## Acknowledgements

This work has been supported in part by grant no. 201/00/1031 of the Grant Agency of the Czech Republic.

## References

1. R.Bělohlávek. *Fuzzy concepts and conceptual structures: induced similarities*. Joint Conf. Inf. Sci.'98 Proceedings, Vol. I, pp. 179-182, Durham, USA, 1998. [Assoc. Intel. Machinery, ISBN 0-9643456-7-6]
2. B. Ganter, R. Wille. *Formal Concept Analysis, Mathematical Foundations*. Springer Verlag 1999. ISBN 3-540-62771-5
3. G.J. Klir, U. St.Clair, B. Yuan. *Fuzzy Set Theory*. Prentice Hall PTR. 1997. ISBN 0-13-341058-7
4. M. Schneider. *Fuzzy Topological Predicates, Their Properties, and Their Integration into Query Languages*. ACM 2001
5. V.Snášel, D. Ďuráková, M. Krátký. *Navigation Through Query Result Using Concept Lattice*. DATSO 2002. ISBN 80-248-0080-2
6. V.Snášel.  $\lambda$ -lattice. *Mathematica Bohemica* No 3., 1997. 267-272.
7. V.Snášel. *Algebraic theory of ordered set*. Submit Algebra Universalis.
8. P.Vojtáš, V.Snášel. *Formal Concept Analysis for Fuzzy Relations*. Submit Kybernetika.
9. R. Wille. *Restructuring lattice theory: an approach based on hierarchies of concepts*. In: I.Rival (ed.): *Ordered sets*. Reidel, Dordrecht-Boston, 445-470. 1982
10. [http://www.mathematik.tu-darmstadt.de/ags/ag1/Literatur/liste\\_intern-en.html](http://www.mathematik.tu-darmstadt.de/ags/ag1/Literatur/liste_intern-en.html)