A Contextual-Logic Extension of TOSCANA

Peter Eklund[1], Bernd Groh[1], Gerd Stumme[2], Rudolf Wille[2]

[1] School of Information Technology, Griffith University, Gold Coast Campus, PMB 50
Gold Coast Mail Centre QLD 9726, Australia; {p.eklund, b.groh}@gu.edu.au
[2] Fachbereich Mathematik , Technische Universität Darmstadt,
Schloßgartenstr. 7, D–64289 Darmstadt;
{stumme,wille}@mathematik.tu-darmstadt.de

Abstract. The aim of this paper is to indicate how TOSCANA may be extended to allow graphical representations not only of concept lattices but also of concept graphs in the sense of Contextual Logic. The contextual-logic extension of TOSCANA requires the logical scaling of conceptual and relational scales for which we propose the Peircean Algebraic Logic as reconstructed by R. W. Burch. As graphical representations we recommend, besides labelled line diagrams of concept lattices and Sowa's diagrams of conceptual graphs, particular information maps for utilizing background knowledge as much as possible. Our considerations are illustrated by a small information system about the domestic flights in Austria.

Contents

1. TOSCANA
2. Set-theoretical Semantics of Contextual Logic
3. Conceptual and Relational Scales
4. Graphical Representations

1 TOSCANA

TOSCANA has been developed at Darmstadt University of Technology as a computer program for analyzing and exploring data by methods of *Formal Concept Analysis* (see [KSVW94], [VW95], [GW99]). It has been used in a wide range of application domains such as medicine, psychology, social sciences, linguistics, information sciences, machine and civil engineering etc. (cf. [GSW98], [SW00]). A typical application combines TOSCANA and a (relational) database to a TOSCANA-*system* allowing the representation, the maintainence, and the activation of information so that users of the system may gain actual knowledge about interesting aspects of the relevant application domain. For the human-machine interaction, TOSCANA offers labelled (nested) line diagrams of concept lattices, representing conceptual relationships of the stored data, and allows the navigation through the data by changing from one line diagram to another (and so on). Although the graphical representations of concept lattices have been proven useful in numerous applications, there are many cases in which such representations of conceptual hierarchies are not sufficient and, in addition, representations of (non-hierarchical) relations become desirable. Those cases have stimulated the extension of Formal Concept Analysis to *Contextual Logic*, a formal logic semantically based on formal contexts from which concept lattices are derivable and also concept graphs, combining formal concepts and conceptual relations (see [Wi97], [Pr98], [PW99]). The aim of this paper is to indicate how TOSCANA may be extended to allow graphical representations not only of concept lattices but also of concept graphs in the sense of Contextual Logic.

Before discussing a contextual-logic extension of TOSCANA, we explain through an example how TOSCANA can be used to establish a conceptual information system based on methods of Formal Concept Analysis. The data for the example, which have already been considered in [PW99], are described in Figure 1 by a data table about the domestic flights in Austria. For a TOSCANA-system, such table is usually considered as a *many-valued context* (G, M, W, I) for which G is the object set comprising all listed flights, M is the attribute set consisting of the attributes "Airline", "Departure Airport", "Departure Time", "Arrival Airport", "Arrival Time", "Days", and "Aircraft", W is a set containing all attribute values described by the entries in the columns of the table, and I is the ternary relation between G, M, and W indicating which object has which value for which attribute.

The first step of establishing a TOSCANA-system for the many-valued context represented in Figure 1 is to turn it into a number of formal contexts, called *conceptual scales*, grasping the information coded by the data. Such transformation should be guided by purposes which we assume in our example to lie in the support of flight information. For our explanations, we choose the seven conceptual scales "Connections", "Departure Time (Hours)", "Departure Time (Minutes)", "Arrival Time (Hours)", "Arrival Time (Minutes)", "Days", and "Airline/Aircraft". The concept lattice of the conceptual scale "Connections" is presented in Figure 2; it yields the information about the departure and arrival airports of the 75 domestic flights which are denoted by their flight number; for instance, the flight 1583 departs from Innsbruck and arrives at Graz (surprisingly, there is no flight from Graz to Innsbruck). The information about the departure time of the flights is well represented by the two interordinal scales "Departure Time (Hours)" and "Departure Time (Minutes)" where the first has the attributes $\leq 6, \ldots, \leq 23$ and $\geq 6, \ldots, \geq 23$ and the second has the attributes $\leq 00, \leq 05, \ldots, \leq 50, \leq 55$ and $\geq 00, \geq 05, \ldots, \geq 50, \geq 55$ (cf. [PW99], Fig. 3); for instance, the departure time 15.10 of flight 1583 is uniquely determined by the attributes ≤ 15 and ≥ 15 of the first scale and ≤ 10 and ≥ 10 by the second scale. The arrival time is represented analogously by two interordinal scales. The choice of interordinal scales for representing time has been well proved because they allow the expression of time intervals by formal concepts. The conceptual scale "Airline/Aircraft" has the attributes "VO", "OS", "F70", "DH8", and "CRJ" yielding a six-element concept lattice. The concept lattice of the conceptual scale "Days", restricted to the flights from Vienna to Innsbruck, is presented in Figure 3.

		Departure		Arrival			
Flight	Airline	Airport	Time	Airport	Time	Days	Aircraft
070	VO	Vienna	07.50	Innsbruck	08.40	1-6	F70
071	VO	Innsbruck	06.25	Vienna	07.20	1-5	F70
072a	VO	Vienna	10.20	Innsbruck	11.35	6	DH8
072b	VO	Vienna	10.50	Innsbruck	12.05	1-5, 7	DH8
073a	VO	Innsbruck	08.35	Vienna	09.45	67	DH8
073b	VO	Innsbruck	09.05	Vienna	09.55	1-5	F70
074	VO	Vienna	13.55	Innsbruck	15.10	2-5, 7	DH8
075	VO	Innsbruck	11.40	Vienna	12.50	1-5	DH8
076a	VO	Vienna	17.45	Innsbruck	18.40	1-6	F70
076b	VO	Vienna	18.40	Innsbruck	19.55	7	DH8
077	VO	Innsbruck	15.35	Vienna	16.45	2-5	DH8
078a	VO	Vienna	20.35	Innsbruck	21.25	1-4	F70
078b	VO	Vienna	21.30	Innsbruck	22.45	7	DH8
078c	VO	Vienna	21.40	Innsbruck	22.35	5	CRJ
330	VO	Linz	06.20	Salzburg	06.50	1-6	CRJ
331	VO	Salzburg	11.20	Linz	11.45	1-5	CRJ
332	VO	Linz	16.05	Salzburg	16.35	1-5	CRJ
333	VO	Salzburg	21.50	Linz	22.15	1-5, 7	CRJ
409	VO	Graz	12.10	Linz	12.45	1-5	CRJ
410	VO	Linz	16.10	Graz	16.50	1-5	CRJ
412	VO	Linz	10.35	Graz	11.10	1-5	CRJ
413	VO	Graz	06.15	Salzburg	06.50	1-5	CRJ
415	VO	Graz	17.30	Salzburg	18.10	1-5	CRJ
416	VO	Salzburg	21.50	Graz	22.25	1-5, 7	CRJ
417	VO	Graz	17.15	Linz	17.45	7	CRJ
501	VO	Klagenfurt	06.00	Salzburg	06.45	1-5	DH8
502	VO	Salzburg	21.55	Klagenfurt	22.40	1-5, 7	DH8
531*	VO-OS	Linz	06.00	Vienna	06.45	1-6	DH8
532*	VO-OS	Vienna	10.40	Linz	11.20	1-5, 7	DH8
533*	VO-OS	Linz	08.35	Vienna	09.25	1-7	DH8
534*	VO-OS	Vienna	22.15	Linz	23.00	1-5, 7	DH8
536a*	VO-OS	Vienna	17.10	Linz	17.55	5	DH8
536b*	VO-OS	Vienna	17.15	Linz	17.55	1-4, 7	DH8
537*	VO-OS	Linz	12.00	Vienna	12.50	1-5, 7	DH8
538*	VO-OS	Vienna	20.30	Linz	21.15	1-7	DH8
539*	VO-OS	Linz	18.15	Vienna	19.00	1-5, 7	DH8
540*	VO-OS	Vienna	10.45	Graz	11.30	1-7	DH8
541*	VO-OS	Graz	06.05	Vienna	06.45	1-6	DH8
542*	VO-OS	Vienna	13.50	Graz	14.35	1-5	DH8
543*	VO-OS	Graz	08.50	Vienna	09.35	1-7	DH8
544*	VO-OS	Vienna	17.20	Graz	18.00	1-7	DH8
545*	VO-OS	Graz	11.55	Vienna	12.35	1-5	DH8
546*	VO-OS	Vienna	19.40	Graz	20.20	1-7	DH8
547*	VO-OS	Graz	15.30	Vienna	16.15	1-5, 7	DH8
548*	VO-OS	Vienna	22.30	Graz	23.10	1-5, 7	DH8
549*	VO-OS	Graz	18.25	Vienna	19.05	1-5, 7	DH8
550*	VO-OS	Vienna	07.25	Klagenfurt	08.15	1-5	DH8
551*	VO-OS	Klagenfurt	06.00	Vienna	06.50	1-6	DH8
552*	VO-OS	Vienna	10.40	Klagenfurt	11.30	1-7	DH8
553*	VO-OS	Klagenfurt	08.40	Vienna	09.30	1-7	DH8
554*	VO-OS	Vienna	13.55	Klagenfurt	14.50	1-5	DH8
555*	VO-OS	Klagenfurt	11.55	Vienna	12.45	1-7	DH8
556*	VO-OS	Vienna	17.10	Klagenfurt	18.00	1-7	DH8
557*	VO-OS	Klagenfurt	15.15	Vienna	16.10	1-5	DH8
558*	VO-OS	Vienna	19.50	Klagenfurt	20.45	1-7	DH8
559*	VO-OS	Klagenfurt	18.20	Vienna	19.10	1-7	DH8
560*	VO-OS	Vienna	22.30	Klagenfurt	23.20	457	DH8
561*	VO-OS	Klagenfurt	21.00	Vienna	22.00	457	DH8
590*	VO-OS	Vienna	10.25	Salzburg	11.20	1-7	DH8
591*	VO-OS	Salzburg	17.15	Vienna	18.10	7	DH8
593*	VO-OS	Salzburg	08.15	Vienna	09.15	1-7	DH8
594*	VO-OS	Vienna	17.35	Salzburg	18.35	1-7	DH8
595*	VO-OS	Salzburg	11.45	Vienna	12.40	1-7	DH8
596a*	VO-OS	Vienna	20.25	Salzburg	21.20	6	DH8
596b*	VO-OS	Vienna	20.35	Salzburg	21.30	1-5, 7	DH8
597*	VO-OS	Salzburg	19.05	Vienna	20.00	1-7	DH8
1557	VO	Klagenfurt	16.00	Vienna	16.50	7	DH8
1583	VO	Innsbruck	15.10	Graz	15.55	7	CRJ
1596	VO	Vienna	14.05	Salzburg	15.05	5	DH8
2980	VO	Innsbruck	06.10	Salzburg	06.40	1-7	DH8
2981	VO	Salzburg	12.30	Innsbruck	13.00	1-7	DH8
2983	VO	Salzburg	16.40	Linz	17.05	1-5	CRJ
2984	VO	Innsbruck	14.35	Salzburg	15.10	1-7	DH8
2985	VO	Salzburg	21.40	Innsbruck	22.05	1-7	DH8
2986	VO	Innsbruck	10.20	Salzburg	10.55	7	DH8

Fig. 1. A data table about the *domestic flights in Austria*

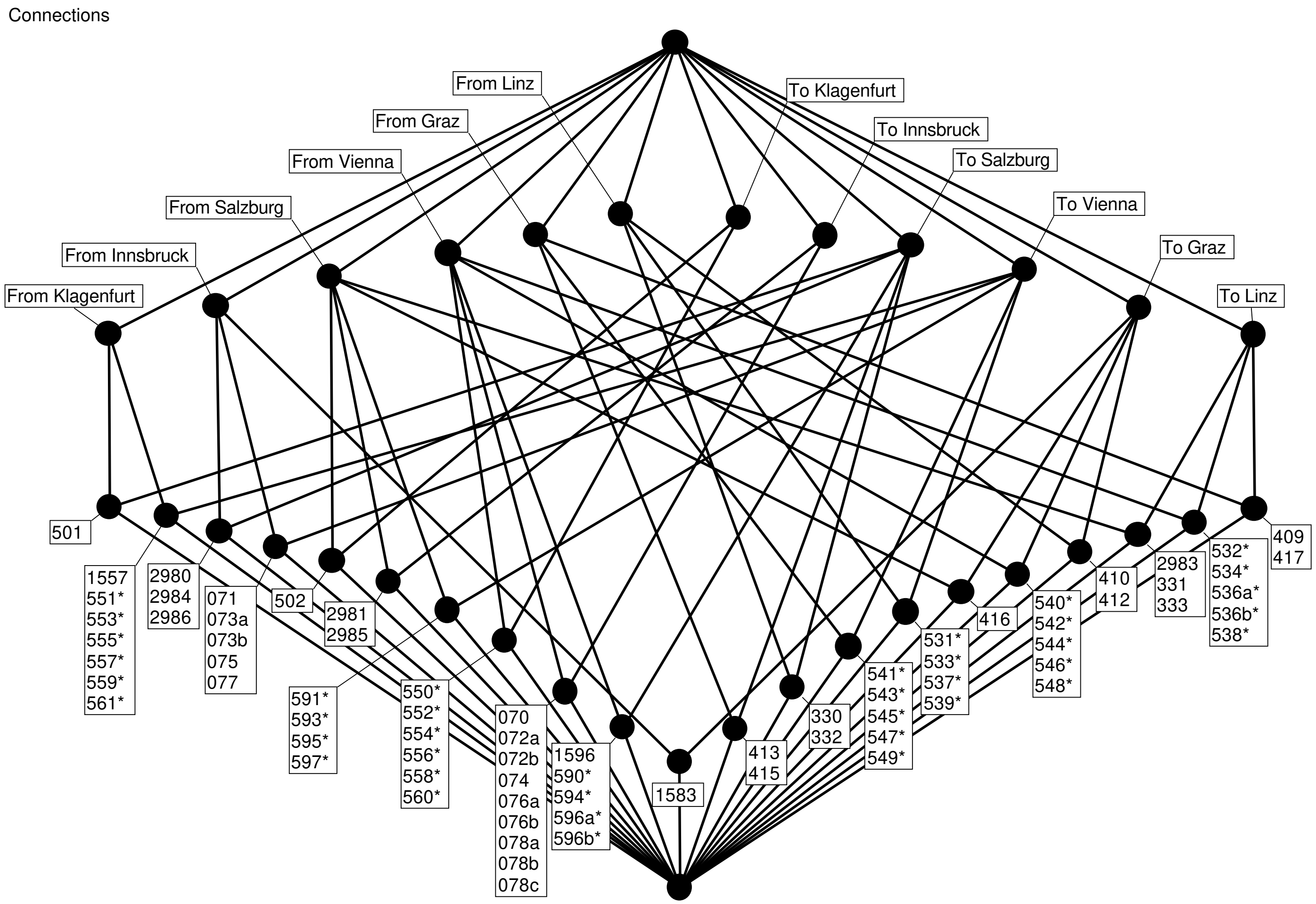

Fig. 2. The concept lattice of the conceptual scale "Connections"

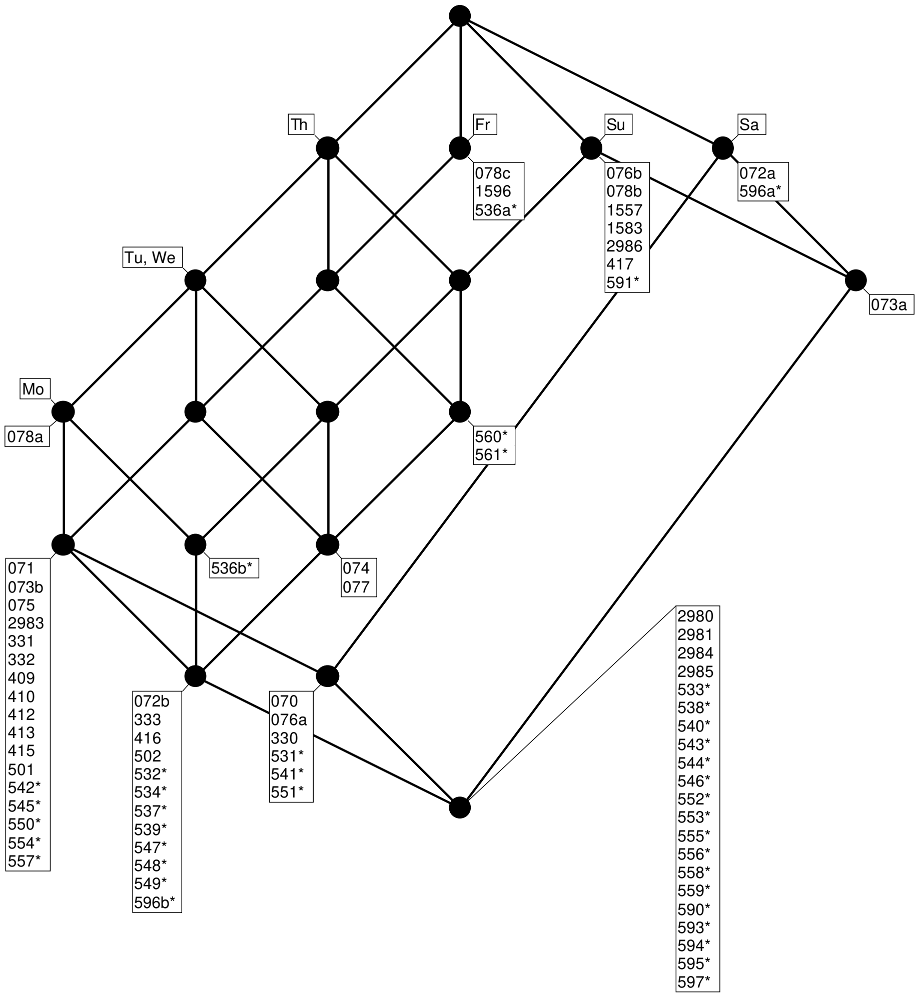

Fig. 3. The concept lattice of the conceptual scale "Days", restricted to the flights from Vienna to Innsbruck

The conceptual scales of a TOSCANA-system, each given by a formal context together with a line diagram of its concept lattice, are viewed on three levels of abstraction to allow high flexibility of their use: First, a conceptual scale is considered as an *abstract scale* having only (clarified) abstract objects and attributes without a particular meaning. Secondly, a conceptual scale is considered as a *concrete scale* having still abstract objects but meaningful attributes with respect to an application domain. Thirdly, a conceptual scale is considered as a *realized scale* having now both: meaningful objects and attributes with respect to an application domain. Abstract scales may be used in different applications,

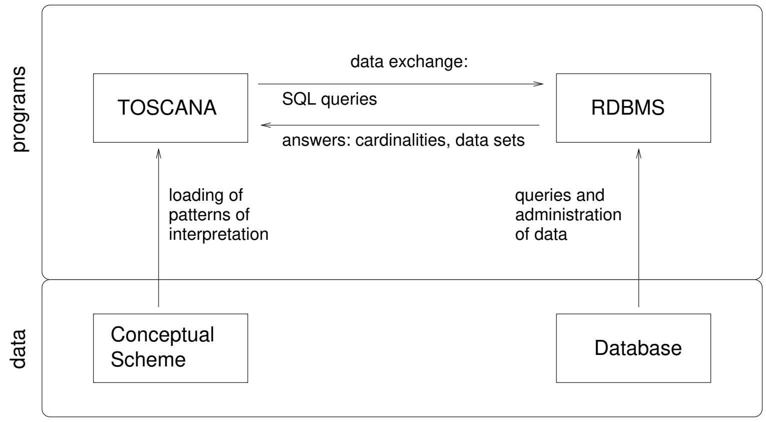

Fig. 4. The basic architecture of a TOSCANA-system

while concrete scales are designed for a specific application, but allow different realizations concerning the set of objects (for instance, concerning actual revisions of the flights in the database on domestic flights in Austria).

The basic architecture of a TOSCANA-system is described in Figure 4. According to that, a TOSCANA-system divides into a data and a program part. In the data part, the information about the objects of the application domain is coded in the (relational) database, while the domain-specific concrete scales (including line diagrams of their concept lattices) and SQL-queries for the objects of the application domain are collected in the *conceptual scheme*; in our example this means that the data table of Figure 1 is stored in the database and that the relevent seven conceptual scales, understood as concrete scales, are available in the conceptual scheme of the data part. For the represention of concrete scales with line diagrams of their concept lattices in the conceptual scheme, the drawing program ANACONDA yields a coding by the description language CONSCRIPT (see [Vo96]). The drawings of the concept lattices has to be provided in advance since, in general, well readable line diagrams cannot be generated automatically.

The program part consists of TOSCANA and a relational database management system (RDBMS). At runtime, TOSCANA loads the conceptual scheme and connects it to a RDBMS which accesses the database in which the information of the application domain is stored. The system offers the user the list of conceptual scales, among which the user may choose one or more scales. If the user chooses one scale, then TOSCANA computes the realized scale and the labelling of the line diagram of its concept lattice by querying the actual objects in the database, and displays the labelled line diagram on the screen. Figure 2, showing the concept lattice of the realized scale “Connections”, is the result of such an action. If the user chooses two (or more) scales, then she can either zoom into a concept of the first scale to obtain a conceptual refinement of this concept by the concept lattice of the second scale, or she can represent the concept lattices of both scales simultaneously by a nested line diagram. Figure 3 results, after choosing the conceptual scales "Connections" and "Days", from zooming in Figure 2 into the ninth object node from the left. A detailed description of TOSCANA-systems can be found in [Na99].

The explanations concerning the TOSCANA-system of the domestic flights in Austria might convince that such an information system could be useful; for instance, the object nodes of Figure 2 partition the flights according to their departure and arrival airports and lead, after drilling down to a suitable object node, further information about departure and arrival time, week days, airline, and even aircraft, relevant to flights from specifically chosen departure and arrival airports. Nevertheless, the representation of all relationships within conceptual hierarchies does not seem sufficient for human-machine communications. Therefore, an extension of TOSCANA should be developed which allows more direct representations of non-hierarchical relations. Such an extension, mathematically based on Contextual Logic, is described in the following sections.

2 Set-theoretical Semantics of Contextual Logic

Contextual Logic has been developed as a mathematization of the traditional philosophical logic with its doctrines of concepts, judgments, and conclusions [Wi00]; for that, the mathematical theory of concepts is taken from Formal Concept Analysis, and the mathematical theory of judgments is derived from Sowa's Theory of Conceptual Graphs [So84]. The set-theoretical semantics of Contextual Logic is based on power context families which are defined as follows: a *power context family* is a sequence $\vec{\mathbb{K}} := (\mathbb{K}_0, \mathbb{K}_1, \ldots, \mathbb{K}_n)$ $(n \geq 2)$ of formal contexts $\mathbb{K}_k := (G_k, M_k, I_k)$ with $G_k \subseteq (G_0)^k$ for $k = 1, \ldots, n$. The formal concepts of $\mathbb{K}_k$ $(1 \leq k \leq n)$ represent by their extents k-ary relations on the object set G_0; they are therefore called the k-ary *relational concepts* of $\vec{\mathbb{K}}$.

According to [PW99], the *formal judgments*, also called *concept graphs*, of a power context family can be represented (up to logical equivalence) as the elements of all the direct products

$$\prod_{(k,g)\in U} [\gamma_k g)$$

for which U is a finite subset of $\bigcup_{k=0,\ldots,n}\{k\} \times G_k$ satisfying the implication

$$(k,g) \in U \text{ and } g = (g_1, \ldots, g_k) \Longrightarrow (0, g_1), \ldots, (0, g_k) \in U \quad (1 \leq k \leq n),$$

and $[\gamma_k g)$ is the principal filter of $\underline{\mathfrak{B}}(\mathbb{K}_k)$ generated by the smallest concept having g in its extent. An element $\vec{\mathfrak{a}} := (\mathfrak{a}_{(k,g)} \mid (k,g) \in U)$ of the product can be understood as the family of the *atomic concept graphs* $[\mathfrak{a}_k : g]$ with $(k,g) \in U$ where the so-called *conceptual instance* $[\mathfrak{a}_0 : g]$ is the element $\vec{\mathfrak{a}}$ of the (one-factor) product with $U := \{(0,g)\}$ and $\mathfrak{a}_{(0,g)} = \mathfrak{a}_0$ and, for $k = 1, \ldots, n$, the so-called *relational instance* $[\mathfrak{a}_k : g]$ is the element $\vec{\mathfrak{a}}$ of the product with $U := \{(k,g), (0,g_1), \ldots, (0,g_k)\}$ (if $g = (g_1, \ldots, g_k)$) and $\mathfrak{a}_{(k,g)} = \mathfrak{a}_k$, $\mathfrak{a}_{(0,g_1)} =$

$\mathfrak{u}_0, \ldots, \mathfrak{a}_{(0,g_k)} = \mathfrak{u}_0$ (in general, $\mathfrak{u}_k$ denotes the largest concept of $\mathbb{K}_k$); instead of $[\mathfrak{a}_k : g]$ we also write $(\mathfrak{a}_k; [\mathfrak{u}_0 : g_1], \ldots, [\mathfrak{u}_0 : g_k])$. Notice that a pair $(\mathfrak{a}_k, g)$ with $\mathfrak{a}_k \in \mathfrak{B}(\mathbb{K}_k)$ and $g \in G_k$ forms an atomic concept graph if and only if $g \in Ext(\mathfrak{a})$.

The formal judgments described in the preceding paragraph may be called *concept graphs in atomistic normal form.* From a concept graph in atomistic normal form one can deduce its equivalent concept graphs using the following equivalences (cf. [PW99], Prop. 1):

(1) $([\mathfrak{a}_t : g] \mid t \in T)$ is equivalent to $[\bigwedge_{t\in T} \mathfrak{a}_t : g]$,
(2) $([\mathfrak{a} : g_t] \mid t \in T)$ is equivalent to $[\mathfrak{a} : \{g_t \mid t \in T\}]$,
(3) $(\mathfrak{a}; [\mathfrak{b}_1 : B_1], \ldots, [\mathfrak{b}_k : B_k])$
is equivalent to $([\mathfrak{a}_k : B_1 \times \cdots \times B_k], [\mathfrak{b}_1 : B_1], \ldots, [\mathfrak{b}_k : B_k])$,
(4) $((\mathfrak{a}; [\mathfrak{b}_1 : B_1], \ldots, [\mathfrak{b}_k : B_k]), (\tilde{\mathfrak{a}}; [\tilde{\mathfrak{b}}_1 : \tilde{B}_1], \ldots, \ldots, [\tilde{\mathfrak{b}}_l : \tilde{B}_l]))$
is equivalent to
$((\mathfrak{a}; [\mathfrak{b}_1 : B_1], \ldots, [\mathfrak{b}_k : B_k]), (\tilde{\mathfrak{a}}; [\tilde{\mathfrak{b}}_1 : \tilde{B}_1], \ldots, [\tilde{\mathfrak{b}}_l : \tilde{B}_l]))/\theta(i,j)$
if $[\mathfrak{b}_i : B_i] = [\tilde{\mathfrak{b}}_j : \tilde{B}_j]$.

The equivalences (1), (2), and (3) are applied to suitable subfamilies of a family of elementary concept graphs to obtain another family of elementary concept graphs (in [PW99], an *elementary concept graph* is understood as a concept graph with at most one relational concept). The equivalence (4) allows us to identify equal conceptual instances related to different relational concepts, i. e., (4) is used to split up a concept graph in elementary concept graphs and to combine concept graphs to one concept graph.

Formal judgments (i.e. concept graphs), which may be easily turned into textual statements of "plain English" (see [ME99]), are useful for expressing specific information contained in a given power context family; they are the means "to make the data talk". Therefore, it is desirable to derive from the data in a relational database a power context family allowing one to deduce informative concept graphs. For doing this, the method of *relational scaling* has been proposed in [PW99]: If a relational database is given in form of Codd's relational model $R \subseteq \prod_{t\in T} A_t$, then a derived power context family $\vec{\mathbb{K}} := (\mathbb{K}_0, \mathbb{K}_1, \ldots, \mathbb{K}_n)$ of formal contexts $\mathbb{K}_k := (G_k, M_k, I_k)$ $(0 \leq k \leq n)$ is formed by suitable subsets $G_0 \subseteq \bigcup_{t\in T} A_t$ and $G_k \subseteq (G_0)^k$ together with corresponding attribute sets giving purpose-oriented meaning to the data (the elements of G_k should be restrictions of elements of $\prod_{t\in T} A_t$).

Understanding the data table in Figure 1 as a representation of a relational model in the sense of Codd, we may derive a power context family $(\mathbb{K}_0, \mathbb{K}_2)$ by the following definitions of object and attribute sets (cf. [PW99]):

$$
\begin{aligned}
G_0 := {} & \{070, 071, \ldots, 2986\} \cup \\
& \{Graz, Innsbruck, Klagenfurt, Linz, Salzburg, Vienna\} \cup \\
& \{06.00, 06.05, \ldots, 23.20\} \cup \\
& \{1, 2, 3, 4, 5, 6, 7\} \\
G_2 := {} & \{(070, Vienna), (071, Innsbruck), \ldots, (2986, Innsbruck)\} \cup \\
& \{(070, Innsbruck), (071, Vienna), \ldots, (2986, Salzburg)\} \cup \\
& \{(070, 07.50), (071, 06.25), \ldots, (2986, 10.20)\} \cup \\
& \{(070, 08.40), (071, 07.20), \ldots, (2986, 10.55)\} \cup \\
& \{(070, 1), \ldots, (070, 6), (071, 1), \ldots (071, 5), \ldots, (2986, 7)\} \\
M_0 := {} & \{flight, airline, airport, time, days, aircraft\} \cup \\
& \{Graz, Innsbruck, Klagenfurt, Linz, Salzburg, Vienna\} \cup \\
& \{\geq 20min, \geq 25min, \geq 30min, \geq 3km, \geq 4km, \geq 5km, \geq 12km, \geq 16km\} \cup \\
& \{\leq 06.00, \leq 06.05, \ldots, \leq 23.20, a.m., p.m.\} \cup \\
& \{\geq 06.00, \geq 06.05, \ldots, \geq 23.20, a.m., p.m.\} \cup \\
& \{Mo, Tu, We, Th, Fr, Sa, Su\} \cup \\
& \{VO, OS, F70, DH8, CRJ\} \\
M_2 := {} & \{From, To, Dept, Arrv, FlDays\}
\end{aligned}
$$

The attributes "$\geq 20min$", "$\geq 25min$","$\geq 30min$" indicate the connecting times at the corresponding airport and the attributes "$\geq 3km$","$\geq 4km$", "$\geq 5km$","$\geq 12km$","$\geq 16km$" indicate the distance from the airport to the corresponding city; furthermore, the attributes "*From*" resp. "*To*" apply to the pairs consisting of a flight number and a departure resp. arrival airport; analogously, the attributes "*Dept*" resp. "*Arrv*" are understood with respect to the departure resp. arrival times. The incidence relations I_0 and I_2 of the formal contexts $\mathbb{K}_0$ and $\mathbb{K}_2$ reflect the declared or obvious meaning of the attributes with respect to the given objects.

3 Conceptual and Relational Scales

In this section we discuss how a contextual-logic extension of a TOSCANA-system may be developed on the basis of a power context family $\vec{\mathbb{K}} := (\mathbb{K}_0, \mathbb{K}_1, \ldots, \mathbb{K}_n)$ of formal contexts $\mathbb{K}_k := (G_k, M_k, I_k)$ $(0 \leq k \leq n)$. It seems most natural that the extended TOSCANA program is designed for the purpose of combining several TOSCANA-systems, one for each formal context $\mathbb{K}_k$; in this setting, the scales for $\mathbb{K}_0$ are called the *conceptual scales*, while the scales for $\mathbb{K}_k$ are called the *k-relational scales* $(k = 1, \ldots, n)$. Of course, the TOSCANA-systems of the formal contexts of a power context family should not be independent because satisfactory information based on the data coded in the power context family has also to combine information out of different formal contexts of the family. This combination can be performed by the method of *logical scaling*, introduced in Formal Concept Analysis by S. Prediger (see [Pr97]). In the case of the power context family $\vec{\mathbb{K}}$, logical scaling is combining attributes out of $\bigcup_{j=0}^{n} M_j$ to obtain new attributes for the formal contexts $\mathbb{K}_k$; as a necessary construction tool, we propose the *term formation* of the *"Peircean Algebraic*

Logic" (PAL) which R. W. Burch created as "an attempt to amalgamate various systems of logic that Peirce developed over his long career" (see [Bu91]).

According to PAL, the basic operations to combine attributes of $\bigcup_{j=0}^{n} M_j$ are the *negation* $\neg$, the *permutations* $\pi \in S_j$, and two so-called *joins*, ι_1 and ι_2, extensionally defined for k-ary attributes m and l-ary attributes $\tilde{m}$ $(k \leq l)$ by

(1) $(\neg m)^{I_k} := G_k \setminus m^{I_k}$,
(2) $(\pi m)^{I_k} := \{(g_{\pi(1)}, \ldots, g_{\pi(k)}) \mid (g_1, \ldots, g_k) \in m^{I_k}\}$,
(3) $(\iota_1 m)^{I_{k-2}} := \{(g_1, \ldots, g_{k-2}) \mid \exists g_{k-1} \exists g_k : (g_1, \ldots, g_k) \in m^{I_k}$ and $g_{k-1} = g_k\}$,
(4) $(\iota_2(m, \tilde{m}))^{I_{k+l-2}} := \{(g_1, \ldots, g_{k-1}, \tilde{g}_2, \ldots, \tilde{g}_l) \mid \exists g_k \exists \tilde{g}_1 : (g_1, \ldots, g_k) \in m^{I_k}$ and $(\tilde{g}_1, \ldots, \tilde{g}_l) \in \tilde{m}^{I_l}$ with $g_k = \tilde{g}_1\}$;

furthermore, the k-ary attributes $\bot_k$, $\top_k$, and id_k with $(\bot_k)^{I_k} := \emptyset$, $(\top_k)^{I_k} := G_k$, and $(id_k)^{I_k} := \{(g_1, \ldots, g_k) \mid g_1 = \cdots = g_k \in G_0\}$ are introduced as constants. In general, the *compound attributes* of the power context family $\vec{\mathbb{K}}$ are derived from the unary attributes of $M_0 \cup M_1$, the k-ary attributes of M_k $(2 \leq k \leq n)$, and the constants by recursively applying the negation $\neg$, the permutations π, or the joins ι_1 or ι_2. For suitably choosing compound attributes, it is useful to know further operations derivable from the basic ones, for instance (cf. [Bu91]):

(5) $(\kappa^i m)^{I_{k+1}} := \{(g_1, \ldots, g_{i-1}, g_i, g_i, g_{i+1}, \ldots, g_k) \mid (g_1, \ldots, g_k) \in m^{I_k}\}$,
(6) $(\delta^i m)^{I_{k-1}} := \{(g_1, \ldots, g_{i-1}, g_{i+1}, \ldots, g_k) \mid (g_1, \ldots, g_k) \in m^{I_k}\}$,
(7) $(\xi^{ij} m)^{I_{k-1}} := \{(g_1, \ldots, g_{j-1}, g_{j+1}, \ldots, g_k) \mid \exists g_j : (g_1, \ldots, g_k) \in m^{I_k}$ and $g_i = g_j\}$,
(8) $(\chi(m, \tilde{m}))^{I_{k+l}} := \{(g_1, \ldots, g_k, \tilde{g}_1, \ldots, \tilde{g}_l) \mid (g_1, \ldots, g_k) \in m^{I_k}$ and $(\tilde{g}_1, \ldots, \tilde{g}_l) \in \tilde{m}^{I_k}\}$;
(9) $(\eta(m, \tilde{m}))^{I_l} := \{(g_1, \ldots, g_l) \mid (g_1, \ldots, g_l) \in \tilde{m}^{I_l}$ with $(g_1, \ldots, g_k) \in m^{I_k}\}$.

The so-called *comma operation* κ^i "is a briliant device that Peirce was using with great skill and effect" ([Bu91], p.76), the operation δ^i is Bernay's *"Streichung"* operation, and the operations ξ^{ij} and χ are called *hook identity* and *product* in [Bu91]. The *insertion operation* η allows us to activate attributes of smaller arity in contexts with objects of greater arity; for instance, $\eta(m, \top_l)$ is a compound attribute of $\mathbb{K}_l$ with the extent $\{(g_1, \ldots, g_l) \in G_l \mid (g_1, \ldots, g_k) \in m^{I_k}\}$.

How the *term formation* by the basic operations of PAL and their derivatives may be used to obtain a TOSCANA-system for a power context family, shall be demonstrated by our example of the domestic flights in Austria. Examples of conceptual scales for the power context family described in Section 2 are given by the labelled line diagrams in Figure 2, 3, and 4 in [PW99]. Figure 5 below shows a nested line diagram combining representations of two binary relational scales. The relational scale of the outer diagram is determined by the four binary attributes *From, To, Dept*, and *Arrv*, while the relational scale of the inner diagram has the binary compound attributes *Graz, Innsbruck, Klagenfurt, Linz, Salzburg, Vienna, a.m.,* and *p.m.* which are formed from unary attributes by using the insertion operation together with $\top_2$ and the conversion π_2 (i.e. extensionally, the exchange of place 1 and 2 in an object pair). The comparison

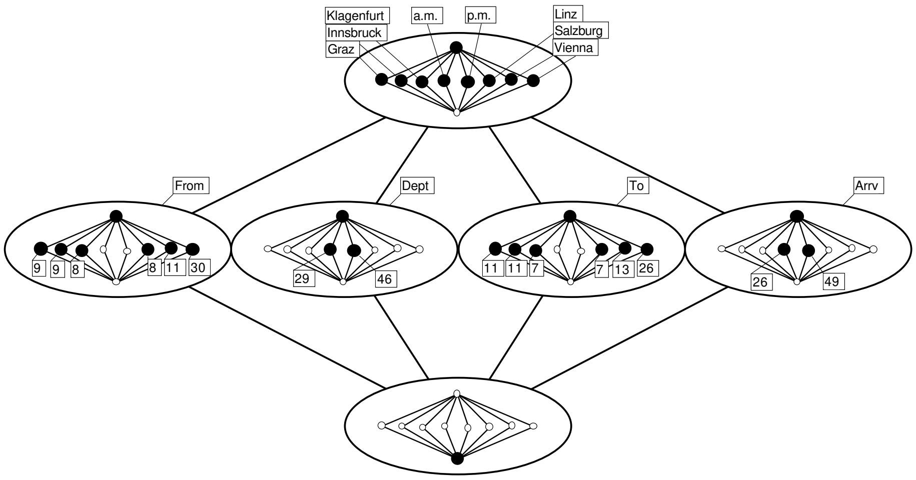

Fig. 5. The concept lattice of the apposition of two binary relational scales

of Figure 5 with Figure 2 makes clear the advantage of using relational scales. Of course, further relational attributes and scales (even of greater arity) are desirable; here we only mention the binary compound attribute $From\text{-}To := \iota_2(\pi_2(From), To)$, the application of which will be shown in the next section.

For using Peircean Algebraic Logic to create compound attributes and conceptual and relational scales based on a power context family, there is not only the argument that this logic combines well with the Contextual Logic as developed until now, but also *Burch's thesis* that "all procedures of relational constructions are formalizable in PAL" ([Bu91], p.122). This thesis is analogous to Church's thesis about computability, of course, less tested, but nevertheless convincing.

4 Graphical Representations

For a TOSCANA-system with a conceptual scheme containing conceptual and relational scales based on a power context family, there is still the question how to *readably represent* concept graphs derived from the conceptual scheme in connection with the actual database. Small concept graphs may be read off from labelled line diagrams of the concept lattices of appositions of relational scales, as the concept graphs about the flight connections between two airports from the nested line diagram in Figure 5. But a little bit larger concept graphs might already diminish the readability seriously. Then, Sowa's conventions for drawing conceptual graphs [So84] would be better for the graphical representation of concept graphs; for our example, such representation is given by Figure 6 in [PW99] showing the flight connections of a commuter between Innsbruck and Vienna.

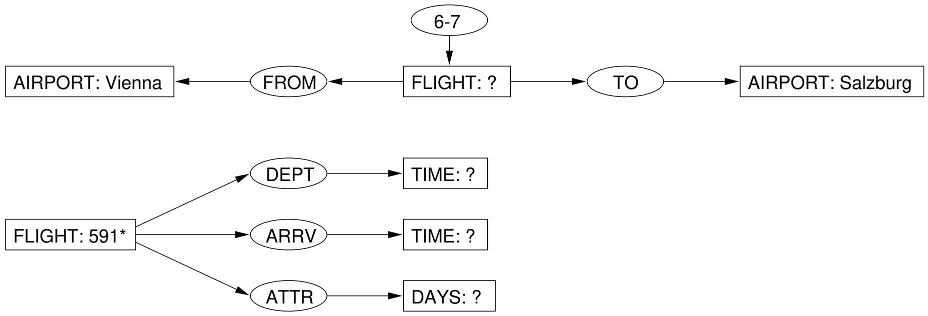

Fig. 6. Query graphs for retrieving flight information

Sowa's graph representations may also be used for formulating *queries* within a TOSCANA-system. This can be performed by choosing from presented line diagrams some conceptual and relational instances and by combining those instances to a graphically represented concept graph in which one or more references are replaced by question marks. The graphical representation of the *query graphs* is needed for checking them syntactically and semantically. Figure 6 shows some examples of querying flight information using concept graphs. In [GE99], it is discussed how those query graphs may be algorithmically tranformed into SQL statements for querying the database of the TOSCANA-system.

Since the intention for developing TOSCANA was always to offer the user rich information in a transparent way so that she can make her decisions based on the presented information which is restricted as little as possible, a TOSCANA-system should allow general queries having as answer a wide "landscape" of detailed facts. For instance, a user, living in Vienna, might ask for the best flight connections at the weekend between Saturday 7 a.m. and Sunday 8 p.m. for visiting Graz, Innsbruck, and Salzburg where she has to meet colleagues to discuss important documents. The full information, under the given restrictions, is represented in the concept graph shown in Figure 7. Obviously, such a representation would be too complex for the customary user. Therefore, we propose another graphical representation, shown in Figure 8, which yields the same flight information as the graph in Figure 7. The easily understood conventions of the shown *information map* are the following: A straight line connecting two towns indicates that there are direct flights between those towns and an arrowhead on such a line points toward their destination; a small table linked to an arrowhead gives the information about the relevant flights, their departure and arrival times, and the week-days they operate. The used graphical means can be described by PAL terms as follows: the arrows represent to the objects of the extent of $From$-To, the link between an arrowhead and a flight number in the relevant table is characterized by the compound attribute $\xi^{13}(\chi(From, To))$, and the columns in the small tables correspond to the objects in the extent of the compound attribute $\eta(\neg\xi^{12}(\neg Sa, \neg Su), \xi^{14}(\chi(\xi^{13}(\chi(Dept, Arrv)), FlDays))$. The descriptions by PAL terms allow a contextual-logic management of the information maps so that they can be integrated in a TOSCANA-system.

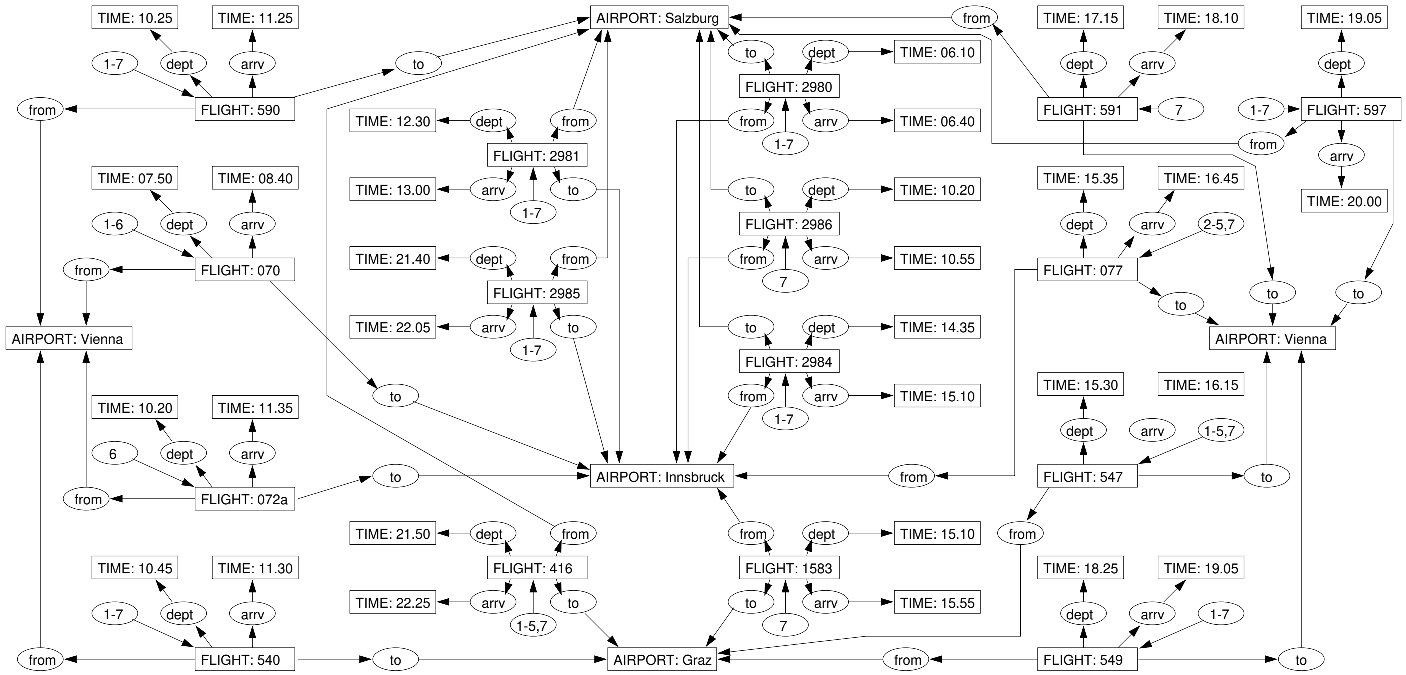

Fig. 7. Conceptual graph representing the weekend flights between Vienna, Graz, Innsbruck, and Salzburg

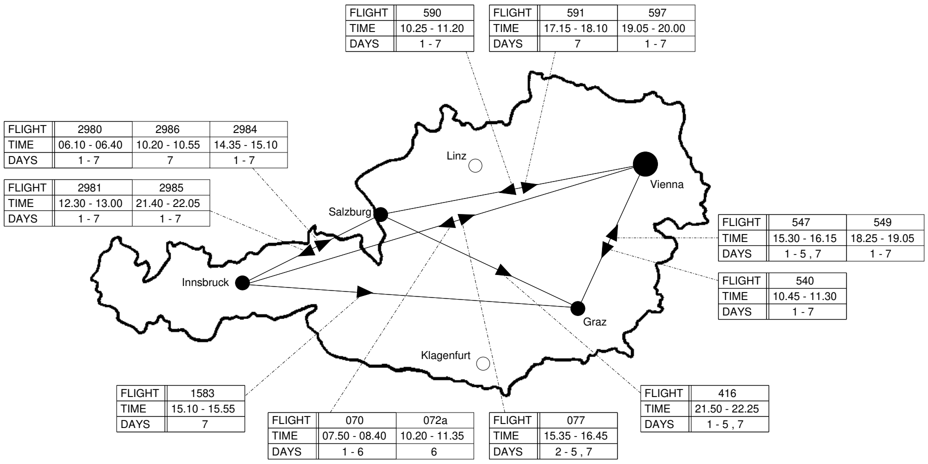

Fig. 8. Information map with the same flight information as the graph in Figure 7

An inspection of the information map immediately shows that tours via Graz to Innsbruck or Klagenfurt are impossible. Therefore, flights to Innsbruck and Salzburg with further connections have to be considered. Since there is only one flight from Salzburg to Graz late on Sunday, the best choices, giving reasonable periods of time for meeting colleagues, seem to be the flights 590 (Vienna-Salzburg), 2985 (Salzburg-Innsbruck), 1583 (Innsbruck-Graz), and 549 (Graz-Vienna); but there is also another solution, namely the flights 070 (Vienna-Innsbruck), 2984 (Innsbruck-Salzburg), 597 (Salzburg-Vienna), 540 (Vienna-Graz), and 549 (Graz-Vienna). Now, the user has to decide, using further information and preferences, for instance, about staying over night in a hotel or at home and about the best possible times for the meetings.

The example teaches us that the development of a TOSCANA-system has to use as much as possible the background knowledge of the potential users, especially, for obtaining a satisfactory human-machine interface. The tremendous increase of readability by changing from Figure 7 to 8 has its explanation in the common knowledge in our present culture, in particular, concerning the use of geographical maps and spatial reasoning. For instance, straight lines between towns with arrowheads on them may be intuitively understood as flight routes, and rows in a table, linked to a flight route, which have two numbers after the word "Time" are identified as the corresponding departure and arrival time without any difficulty (notice that the activated background knowledge has to be made explicit in Figure 7). Of course, the labelled line diagrams of concept lattices are less easily read and require some practice, but our experience is that customers, who are familar with the application domain and are interested in the presented information, understand labelled line diagram astonishingly well. Therefore, also in the future, TOSCANA-systems will use labelled line diagrams for presenting information, but will combine them with information maps and other graphical representations.

Acknowledgement: This paper has substantially benefited from the financial support given by the Australian Research Council (IREX-Programme) which allowed the fourth author to work for two months in 1999 at the School of Information Technology of the Griffith University (Gold Coast Campus).

References

[Bu91] R. W. Burch: *A Peircean reduction thesis*. Texas Tech University Press, Lubbock 1991.

[GW99] B. Ganter, R. Wille: *Formal Concept Analysis: Mathematical Foundations*. Springer, Heidelberg 1999. (Translation of: *Formale Begriffsanalyse: Mathematische Grundlagen*. Springer, Heidelberg 1996.)

[GE99] B. Groh, P. Eklund: Algorithms for creating relational power context families from conceptual graphs. In: W. Tepfenhart, W. Cyre (eds.): *Conceptual Structures: Standards and Practices*. LNAI **1640**. Springer, Heidelberg 1999, 389–400.

[GSW98] B. Groh, S. Strahringer, R. Wille: TOSCANA-systems based on thesauri. In: M.-L. Mugnier, M. Chein (eds.): *Conceptual Structures: Theory, Tools and Applications*. LNAI **1453**. Springer, Heidelberg 1998, 127–138.

[KSVW94] W. Kollewe, M. Skorsky, F. Vogt, R. Wille: TOSCANA – ein Werkzeug zur begrifflichen Analyse und Erkundung von Daten. In: R. Wille, M. Zickwolff (eds.): *Begriffliche Wissensverarbeitung – Grundfragen und Aufgaben*. B. I.-Wissenschaftsverlag, Mannheim 1994, 267–288.

[ME99] Ph. Martin, P. Eklund: Embedding knowledge in Web documents. *Computer Networks* **31** (1999), 1403–1409.

[Na99] NaviCon: TOSCANA 2.1 – Benutzerhandbuch. NaviCon GmbH, Frankfurt 1999.

[Pr97] S. Prediger: Logical scaling in Formal Concept Analysis. In: D. Lukose, H. Delugach, M. Keeler, L. Searle, J. F. Sowa (eds.): *Conceptual structures: fulfilling Peirce's dream*. LNAI **1257**. Springer, Heidelberg 1997, 332–341.

[Pr98] S. Prediger: *Kontextuelle Urteilslogik mit Begriffsgraphen. Ein Beitrag zur Restrukturierung der mathematischen Logik*. Dissertation, TU Darmstadt. Shaker Verlag, Aachen 1998.

[PW99] S. Prediger, R. Wille: The lattice of concept graphs of a relationally scaled context. In: W. Tepfenhart, W. Cyre (eds.): *Conceptual Structures: Standards and Practices*. LNAI **1640**. Springer, Heidelberg 1999, 401–414.

[So84] J. F. Sowa: *Conceptual structures: Information processing in mind and machine*. Adison-Wesley, Reading 1984.

[SW00] G. Stumme, R. Wille (eds.): *Begriffliche Wissensverarbeitung: Methoden und Anwendugen*. Springer, Heidelberg 2000.

[Vo96] F. Vogt: *Datenstrukturen und Algorithmen zur Formalen Begriffsanalyse: Eine C++-Klassenbibliothek*. Springer, Heidelberg 1996.

[VW95] F. Vogt, R. Wille: TOSCANA – A graphical tool for analyzing and exploring data. In: R. Tamassia, I. G. Tollis (eds.): *GraphDrawing '94*. LNCS **894**. Springer, Heidelberg 1995, 226–233.

[Wi97] R. Wille: Conceptual Graphs and Formal Concept Analysis. In: D. Lukose, H. Delugach, M. Keeler, L. Searle, J. F. Sowa (eds.): *Conceptual Structures: Fulfilling Peirce's Dream*. LNAI **1257**. Springer, Heidelberg 1997, 290–303.

[Wi00] R. Wille: *Contextual logic summary*. FB4-Preprint, TU Darmstadt 2000.