Building Sub-Knowledge Bases Using Concept Lattices

J. J. SARBO

Computing Science Institute, University of Nijmegen, Toernooiveld 1, 6525 ED Nijmegen, The Netherlands
Email: janos@cs.kun.nl

A theory of concept (Galois) lattices was first introduced by Wille. An extension of his work to simple structures called concept sublattices has also been published. This paper shows that concept sublattices can be applied to (i) determining subsumption of specifications and (ii) decomposing specifications in terms of others. I show that the latter application of the theory may provide us with new conceptualizations of a specification.

Received May 21, 1996; revised June 9, 1997

1. INTRODUCTION

Consider the following problem: suppose we want to specify notions such as player, card-game, bridge, and describe each of them in terms of a set of primitives, for example, person, seating, etc. So, for example, a kind of player is specified as a person who plays a game against his neighbours (the persons seated next to him) and a sort of card-game as a game played by four persons seated around a table. If we wish to use this specification and determine the relationship between the specified notions, for example whether the card-game has players, then we find ourselves in an inconvenient situation. Although all necessary information is present in the specification, it is given in terms of persons, topology, for example, and therefore not immediately applicable.

A simple-minded solution would, for instance, either allow the use of the notion of the player in the specification of the card-game (which is not the solution we are looking for) or extend the specification of the card-game and add the conditions for person, topology, for example, to constitute a player. By doing this we add, in fact, a copy of the specification of the player to the one of the card-game.

This paper shows that the use of concept lattices allows a more elegant solution to this problem that needs no extra specification. Clearly, the above specification problem is just an example; the method proposed in this paper can be used for a wide range of problems where some structure is to be 'found' as part of another one. It is also shown that by determining such a subsumption relation on specifications we may find decompositions, or even new conceptualizations (sub-knowledge bases) of a specification.

Decomposition of concept lattices is found in [1]. It should be emphasized that the subdirect decomposition of [1] differs from the one described in this paper, as decomposition of concept lattices in terms of a set of given lattices is investigated and embedding of contexts is allowed (see Section 5).

Batch and incremental algorithms for building concept lattices can be found in [2] and [3]. Automatic discovery of implication rules from data using concept lattices is described by Godin and Missaoui [4]. Their approach is close to mine, as implication rules can be used for the characterization of relations and concept lattices. The generation of sub-knowledge bases, however, needs further generalizations.

The structure of the rest of paper is as follows. Section 2 introduces concept lattices; Section 3 recalls basic definitions and theorems of lattice theory. The terminology and much of the presentation is borrowed, with appropriate adjustments, from [5]. Section 4 recapitulates the basic notions and a fundamental theorem on concept lattices. Sections 5 and 6 describe a generalization of the theory and its application. Section 7 briefly summarizes an algorithm for lattice subsumption, and in Section 8 I compare my approach with one using Prolog.

2. CONCEPT LATTICES

Concept lattices were first defined by Wille [6] who introduced them for the formal representation of the philosophical notions of concept and concept hierarchy. Traditionally, the notion of a concept is determined by its extent and its intent, where the extent consists of all objects that share all attributes of the intent and the intent covers all attributes common for all objects of the extent.

Basically a concept lattice is a representation of a (e.g. binary) relation R between a set of objects G (Gegenstände) and a set of attributes M (Merkmale). The triple (G, M, R) is called a context. The lattice arises from that context by applying a Galois connection between the power sets of G and M. This Galois connection is a particular one, called the polar [7].

Various applications of concept lattices have been reported in the literature [8, 3, 4, 9]. Typically, such an application assumes a subset of G (respectively M) to be given as input for which a corresponding subset of M (respectively G) is computed (if such exists) by searching the concept lattice (knowledge base) for a concept having the smallest extent containing the given subset of G. So, for example, given input $\mathsf{A}' \subseteq G$ the answer is a concept (A, B) of the concept lattice, such that $\mathsf{A}' \subseteq \mathsf{A}$ and A is 'smallest'. Then B is the corresponding subset of M.

In this sense a concept lattice is an appropriate representation for what is referred to in artificial intelligence (AI) as 'cooperative communication'[1].

Motivated by different practical problems of visual input processing, recognition of continuous speech and knowledge representation, an extension to the above model was introduced in [10, 12]. The essence of this extension is that a set of interrelated subsets of objects and attributes are allowed to be given as input. It turns out that in this case the input itself can also be represented by means of a Galois connection. Eventually this generalization allows the input to be a concept lattice.

For such an input, similar to the previous example, the answer is the 'smallest' sublattice of the concept lattice (knowledge base) that subsumes it. In summary, we have that an embedding relation for two concept lattices has to be determined (the task of finding a 'smallest' such lattice can be handled separately).

Let us denote those lattices by L_1 and L_2 and their subsumption problem by $L_2 \subseteq L_1$. In general, we may have any number of such L_2 lattices (knowledge bases). We will show that in this case, and under certain conditions, the subsumption problem 'becomes' a decomposition problem for concept lattices which involves the recognition of the given L_2 concept lattice(s) as sublattice(s) of the given concept lattice L_1. Whenever such a decomposition is possible we may obtain an abstract conceptualization of a specification.

An interesting question one may ask is whether the concept lattice construction could be repeatedly applied, in turn to the resulting structure from such a decomposition. The answer is, somewhat unexpectedly, affirmative and in this paper I will show the various possible ways for a recursive application.

Let us make our last statement more precise. Assume that having applied our Galois connection, we constructed a concept lattice. A sufficient condition for the (recursive) application of the Galois connection is that a set of objects and attributes and a relation between them are given, for now, in the concept lattice. Recall that these three components are parameters of the Galois connection, and can arbitrarily be chosen.

A first choice can be based on the trivial observation that a concept lattice is, by definition, an ordered set. We may consider an element of such a lattice as object and its 'sons' as its attributes. Another choice, a generalization of the previous one which seems more practical, takes the sublattices of a concept lattice as objects. In this case the attributes of a sublattice can be defined by a covering relation on sublattices (see Section 6).

[1]In a nutshell, cooperative communication ([11]) is characterized by four maxims of which this representation may help to realize the first two: (i) (maxim of quantity) make your contribution as informative as required, and not more than that, (ii) (maxim of quality) try to make your contribution one that is true, (iii) (maxim of relation) make your contribution relevant with respect to the previous one, (iv) (maxim of manner) be perspicuous.

3. BASIC DEFINITIONS

This section includes some useful definitions and theorems about Galois connections. As indicated above, we make extensive use of [5] and prove 'borrowed' theorems only when they are not proved there. Another source of savings is due to duality: the dual of a theorem may be stated without proving it.

In the following A and B are taken to be sets, A and B to be set variables, and $\mathcal{A} = (A, \sqsubseteq_A)$ and $\mathcal{B} = (B, \sqsubseteq_B)$ to be posets (partially ordered sets). When this will not lead to confusion the subscripts are dropped from the orderings. Let $F_1 \in A \leftarrow B$ and $F_2 \in B \leftarrow A$ be functions.

We denote by $P.x$ the fact that the predicate P might depend on x. For predicates P and Q, that might depend on x, universal quantification, 'for all x such that $P(x)$ holds, $Q(x)$ holds', is written $\forall(x : P.x : Q.x)$. The same formula without quantification and for Q a function denotes set abstraction.

We allow functions to be lifted from elements to sets. If $F : B \leftarrow A$, then for $\mathsf{A} \subseteq A$ the lifted function F is defined by $F.\mathsf{A} = (a : a \in \mathsf{A} : F.a)$.

DEFINITION 3.1. *Galois connection. A pair of functions* (F_1, F_2) *is called a Galois connection iff*

$$\forall(x, y : x \in B \wedge y \in A : F_1.x \sqsubseteq y \equiv x \sqsubseteq F_2.y).$$

If $A = B$ *and* $F_1 = F_2$ *then the Galois connection is called homogeneous.*

THEOREM 3.1. *Let* (F_1, F_2) *be a Galois connection. If* $\mathcal{A}$ *or* $\mathcal{B}$ *are complete lattices then* $F_1.\mathcal{B}$ *and* $F_2.\mathcal{A}$ *are isomorphic complete lattices.*

Later Theorem 3.1 will be applied to the particular case $\mathcal{A} = (\mathcal{P}.A, \subseteq)$ and $\mathcal{B} = (\mathcal{P}.B, \supseteq)$ where A and B denote a set of objects and a set of attributes, respectively, and $\mathcal{P}$ the set-valued function 'power set'. This function is well-defined because the sets to be considered are finite.

It was previously mentioned that polars [7] play a central role in concept lattices. We will define polars stepwise, starting from a set of more simple functions.

DEFINITION 3.2. *For a relation* $R \subseteq A \times B$ *we define a function to* $\mathcal{P}.B$ *from* A *by defining for every* $a \in A$*:* $a.R = \{b : b \in B : aRb\}$.

DEFINITION 3.3. *For a relation* $R \subseteq A \times B$ *we define a function to* $\mathcal{P}.A$ *from* B *by defining for every* $b \in B$*:* $R.b = \{a : a \in A : aRb\}$.

The functions $a.R$ and $R.b$ can be lifted to functions to $\mathcal{P}.B$ from $\mathcal{P}.A$, and to $\mathcal{P}.A$ from $\mathcal{P}.B$, respectively.

DEFINITION 3.4. For every $A \in \mathcal{P}.A$ we define the right polar:

$$\{A\}R = \cap.(a : a \in A : a.R).$$

DEFINITION 3.5. *For every* $B \in \mathcal{P}.B$ *we define the left polar:*

$$R\{B\} = \cap.(b : b \in B : R.b).$$

In order to make the formulae more readable the right resp. left polar of a set A will sometimes be denoted by $R^r.A$ resp. $R^l.A$. If $A = \{a\}$ is a singleton set we simply write $R^r.a$ and $R^l.a$.

THEOREM 3.2. *The pair of polars* (R^r, R^l) *is a Galois connection.*

A proof of this theorem is found in [5]. Birkhoff [7] applies this theorem as a definition for Galois connections.

4. FORMAL CONCEPTS

A concept lattice is yielded by a Galois connection. The 'ingredients' of a Galois connection are a pair of posets and a pair of functions. In the case of concept lattices, these functions are the polars, and the pair of posets are the power sets of the finite sets G and M. The sets G and M are interrelated by the relation R (i.e. $R \subseteq G \times M$).

Formally, the triple (G, M, R) is called the context. For $g \in G, m \in M, (g, m) \in R$ iff object g has the attribute m. A generalization of R to an n-ary relation is found in [6].

DEFINITION 4.1. *For a context the following mappings are defined in [6]:*

a $A' = \{m \in M \mid gRm \text{ for all } g \in A\}$ *for* $A \subseteq G$,
b $B' = \{g \in G \mid gRm \text{ for all } m \in B\}$ *for* $B \subseteq M$.

It is not difficult to recognize that these functions are just the functions from Definitions 3.4 and 3.5 in disguise, that is, $A' = \{A\}R$ and $B' = R\{B\}$. The importance of this fact is underlined by the following theorem.

THEOREM 4.1. For $R \subseteq G \times M$, $G \in \mathcal{P}.G$ and $M \in \mathcal{P}.M$ the functions $\{\_\}R$ and $R\{\_\}$ are a Galois connection by the following equivalence:

$$\{G\}R \supseteq M \equiv G \subseteq R\{M\}.$$

Proof. This is a consequence of Definition 3.1 by the substitution $\{\_\}R$ for F_2 and $R\{\_\}$ for F_1. □

In the theory of concept lattices polars can be interpreted as follows: for $G \in \mathcal{P}.G$, $\{G\}R$ is the set of those attributes that are common for all objects in G. Similarly, for $M \in \mathcal{P}.M$, $R\{M\}$ contains those objects that are common for all attributes in M.

DEFINITION 4.2. *Concept. A concept is a pair* (G, M) *with* $G = R\{M\} = R^l.M$ *and* $M = \{G\}R = R^r.G$.

A concept (G, M) is a (Galois) closed element, as by substitution we get $G = R^l.M = R^l.R^r.G$, and dually $M = R^r.G = R^r.R^l.M$.

DEFINITION 4.3. *Concept lattice. The concept lattice (Begriffsverband) of a context* (G, M, R), *denoted as* $\underline{B}(G, M, R)$, *is the set of concepts with the ordering:*

$$(G_1, M_1) \leq (G_2, M_2) \equiv G_1 \subseteq G_2.$$

THEOREM 4.2. $G_1 \subseteq G_2 \equiv M_1 \supseteq M_2$.

Proof. $G_1 \subseteq G_2$
$\equiv$ { concept }
$G_1 \subseteq R\{M_2\}$
$\equiv$ { Theorem 4.1 }
$\{G_1\}R \supseteq M_2$
$\equiv$ { concept }
$M_1 \supseteq M_2$. □

The concept lattice construction is illustrated by two examples.

EXAMPLE 1. Player, card-game.

The first example is the 'player'. The player is specified as a person who plays a game against his neighbours on the left- and right-hand sides. It is assumed that the specification is 'translated' (by a connoisseur) to the context (G_p, M_p, R_p), where $G_p = \{P_a, P_b, P_c\}$ (P_a denotes the person playing a game, whereas P_b and P_c are his neighbours), $M_p = \{l, r\}$ (l and r are short for 'left' and 'right neighbours') and $R_p = \{(P_a, l), (P_a, r), (P_b, l), (P_c, r)\}$.

Two remarks are in order here. First, the above specification of the player reflects the conception of this notion; obviously this specification is only one of the many possible ones. Secondly, the translation of an informal specification to a context may be complicated. This problem, however, is not treated in this paper. After all, we are only interested in subsumption properties of contexts, or equivalently, their corresponding concept lattices.

According to Theorem 4.1 each subset A of G_p has to be considered and the corresponding subset $\{A\}R_p$ of M_p must be computed. Let us begin with the subset $A = \{P_a\}$. Since A is a singleton set, we find that $\{A\}R_p = \{l, r\}$ (the set of attributes of P_a). Obviously, the set of objects that share all attributes from the set $\{l, r\}$ is the singleton set $\{P_a\}$, as follows from the calculations: $R_p\{\{l, r\}\} = R_p\{\{l\}\} \cap R_p\{\{r\}\} = \{P_a, P_b\} \cap \{P_a, P_c\} = \{P_a\}$. We may conclude that $(\{P_a\}, \{l, r\})$ is a concept.

Consider another subset, say, $A = \{P_a, P_b\}$. By definition, $\{A\}R_p = \{\{P_a\}\}R_p \cap \{\{P_b\}\}R_p = \{l, r\} \cap \{l\} = \{l\}$. Again, $(\{P_a, P_b\}, \{l\})$ is a concept, because $R_p\{\{l\}\} = \{P_a, P_b\}$. The computation of the remaining concepts is straightforward and left to the reader.

Finally, the members of the concept lattice ($\underline{B}_p$) are:

$$C_0 = (\{P_a\}, \{l, r\}),$$
$$C_1 = (\{P_a, P_b, P_c\}, \{\ \}),$$
$$C_2 = (\{P_a, P_b\}, \{l\}),$$
$$C_3 = (\{P_a, P_c\}, \{r\}).$$

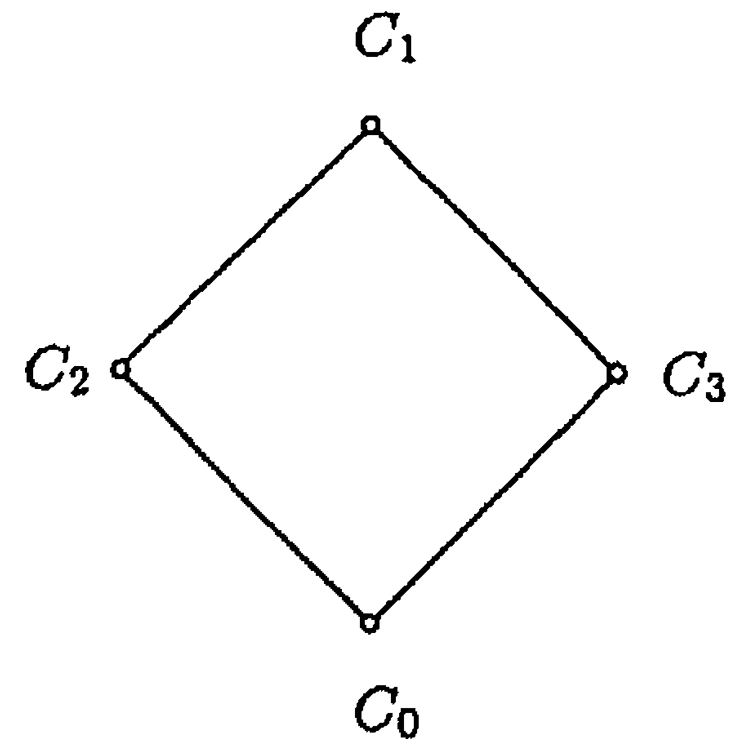

FIGURE 1. Concept lattice of the player.

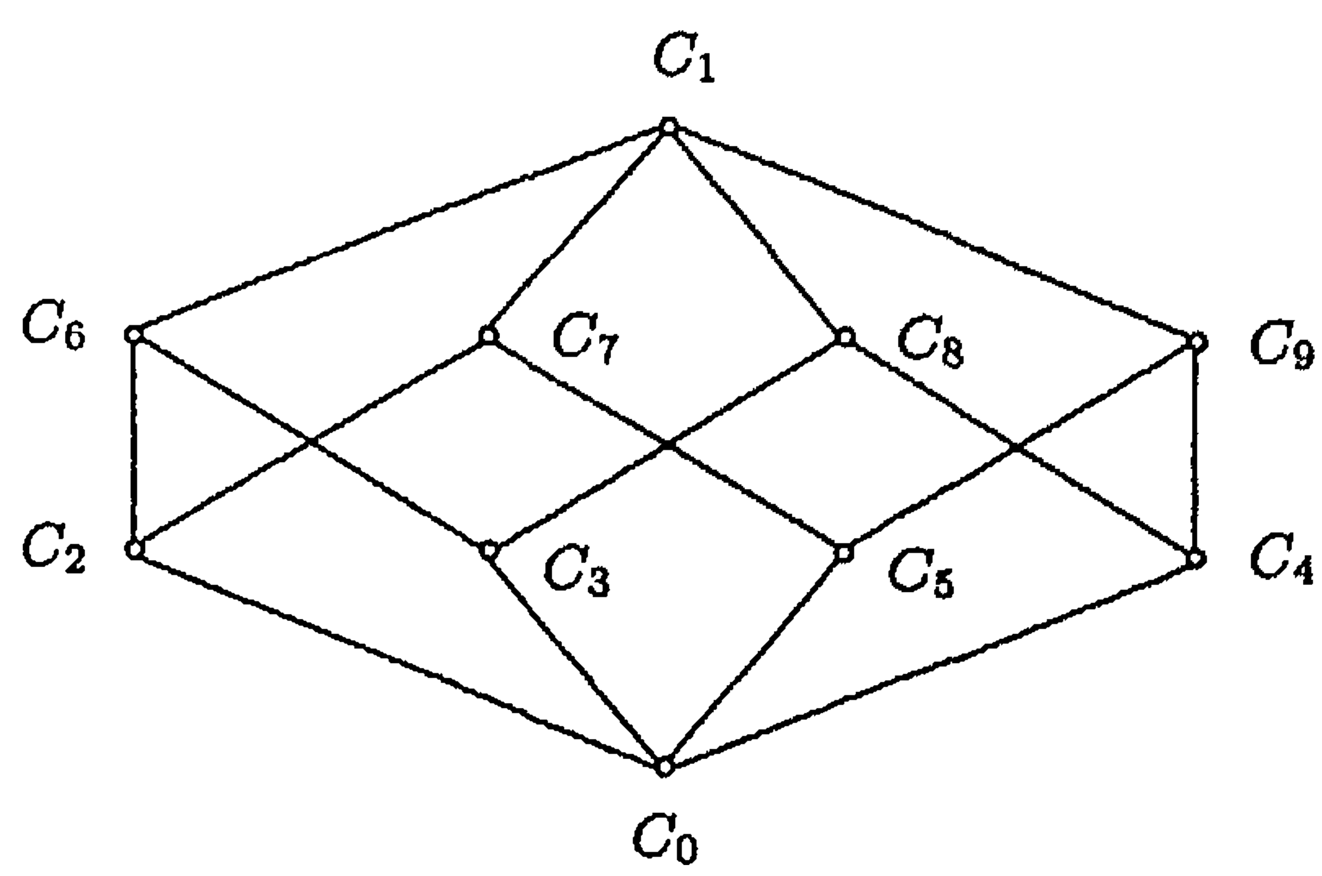

FIGURE 2. Concept lattice of the card-game.

The lattice is depicted by its Hasse diagram in Figure 1.

The second example is the 'card-game'. We specify it as a game played by four persons P_0–P_3, seated at four sides of a table, s_0–s_3. Following the same procedure as above, we obtain the context (G_g, M_g, R_g), where $G_g = \{P_0, P_1, P_2, P_3\}$, $M_g = \{s_0, s_1, s_2, s_3\}$ and $R_g = \{(P_0, s_0), (P_0, s_1), (P_1, s_1), (P_1, s_2), (P_2, s_2), (P_2, s_3), (P_3, s_0), (P_3, s_3)\}$.

The concept lattice $(\underline{\mathcal{B}}_g)$, shown in Figure 2, has the following elements:

$$\begin{aligned}
C_0 &= (\{\,\}, \{s_0, s_1, s_2, s_3\}),\\
C_1 &= (\{P_0, P_1, P_2, P_3\}, \{\,\}),\\
C_2 &= (\{P_0\}, \{s_0, s_1\}),\\
C_3 &= (\{P_1\}, \{s_1, s_2\}),\\
C_4 &= (\{P_2\}, \{s_2, s_3\}),\\
C_5 &= (\{P_3\}, \{s_0, s_3\}),\\
C_6 &= (\{P_0, P_1\}, \{s_1\}),\\
C_7 &= (\{P_0, P_3\}, \{s_0\}),\\
C_8 &= (\{P_1, P_2\}, \{s_2\}),\\
C_9 &= (\{P_2, P_3\}, \{s_3\}).
\end{aligned}$$

Now recall the fundamental theorem on concept lattices from [6].

THEOREM 4.3. *Part 1. Let (G, M, R) be a context. Then $\underline{\mathcal{B}}(G, M, R)$ is a complete lattice in which infimum and supremum can be described as follows:*

a $\bigsqcup_{j \in J} (A_j, B_j) = ((R^l.R^r. \bigcup_{j \in J} A_j), \bigcap_{j \in J} B_j)$,

b $\bigsqcap_{j \in J} (A_j, B_j) = (\bigcap_{j \in J} A_j, R^r.R^l.(\bigcup_{j \in J} B_j))$.

Part 2. A complete lattice L is isomorphic to $\underline{\mathcal{B}}(G, M, R)$ iff there are mappings $\gamma \in L \leftarrow G$ and $\mu \in L \leftarrow M$ such that $\gamma.G$ is join-dense in L (i.e. $L = \{\sqcup X \mid X \subseteq \gamma.G\}$), $\mu.M$ is meet-dense in L (i.e. $L = \{\sqcap X \mid X \subseteq \mu.M\}$), and $gRm \equiv \gamma.g \le \mu.m$ for all $g \in G$ and $m \in M$.

Informally, part 1 of the theorem states that, for a set of concepts the infimum (or join) is a concept that can be computed as follows: take the union of all of the first components of the set of concepts. This yields a set of objects $\bigcup_{j \in J} A_j$. Then, determine for this set the (Galois) closed elements, that is, $R^l.R^r. \bigcup_{j \in J} A_j$ which is a superset of $\bigcup_{j \in J} A_j$. Clearly, those objects of $\bigcup_{j \in J} A_j$ will occur in the closure that have some attribute in common with all other objects of the set. Those 'common' attributes are given by $\bigcap_{j \in J} B_j$.

Part 2 is about the construction of a concept lattice. It states that some elements of the concept lattice can be computed by the functions γ and μ exclusively from G and M, and all other elements are the join (or meet) of those elements. This latter condition is expressed in the theorem by demanding denseness of L.

5. SUBSUMPTION RELATION

Let us return to the problem mentioned in the Introduction. Essentially, the problem is about a subsumption relation on specifications. In this section it is shown that the concept lattice representation of specifications offers an elegant solution to that problem. The idea is that (under certain conditions) the subsumption relation on specifications can be 'translated' into the sublattice relation on concept lattices.

DEFINITION 5.1. *A sublattice of a lattice L is a non-empty subset X of L, such that $a \in X$ and $b \in X$ imply $a \sqcup b \in X$ and $a \sqcap b \in X$.*

The present definition of a sublattice of a concept lattice is based on the definition of sublattice above. Since an element of such a lattice is a pair of sets, it is necessary that a concept lattice is a sublattice of another one if it is a sublattice (in the above sense) and a consistent mapping of corresponding elements of the two lattices exists. This condition is expressed by demanding that the diagram of Definition 5.5 commutes.

DEFINITION 5.2. *A lattice L_1 is homomorphic to a lattice L_2 if there exists an injective, order-preserving map h from L_2 to L_1. Then L_1 is homomorphic to L_2 by h. If h is bijective, it is called an isomorphism.*

DEFINITION 5.3. *Let $\underline{\mathcal{B}}_1 = \underline{\mathcal{B}}(G_1, M_1, R_1)$ and $\underline{\mathcal{B}}_2 = \underline{\mathcal{B}}(G_2, M_2, R_2)$ be concept lattices. Let us denote their bottom and top elements, for $i = 1, 2$, by $\bot_i = (G_i^{\bot}, M_i^{\bot})$ and $\top_i = (G_i^{\top}, M_i^{\top})$. We say $\underline{\mathcal{B}}_2$ is compatible with $\underline{\mathcal{B}}_1$ if $G_2^{\bot} \subseteq G_1^{\bot}$ and $M_2^{\top} \subseteq M_1^{\top}$.*

DEFINITION 5.4. *A mapping between contexts, $\varphi \in (G_1, M_1, R_1) \leftarrow (G_2, M_2, R_2)$, is called a context*

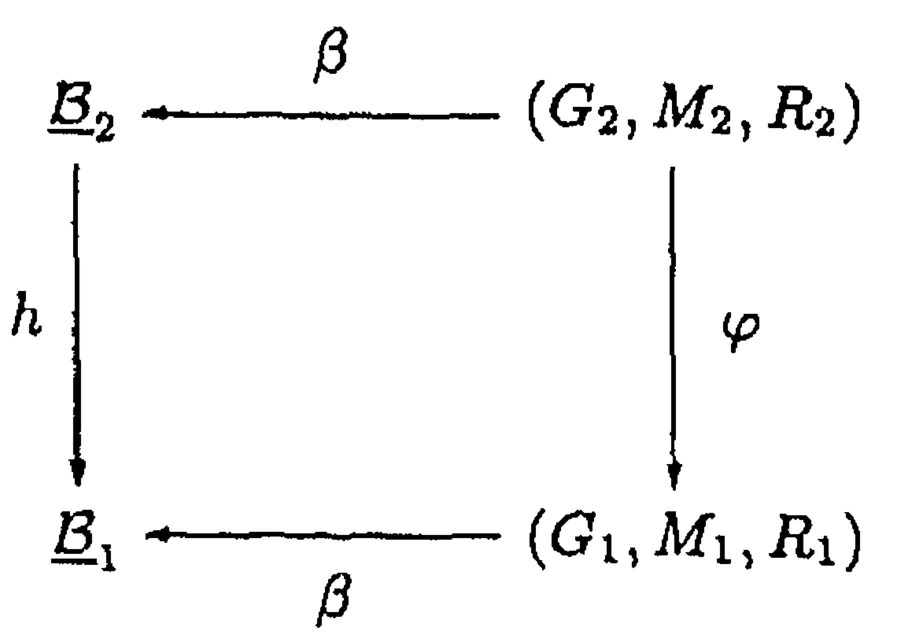

FIGURE 3. Concept sublattice.

embedding if $\varphi = (\varphi_G, \varphi_M)$ *is a pair of injective maps,* $\varphi_G.G_2 \subseteq G_1$, $\varphi_M.M_2 \subseteq M_1$ *and* $(\varphi_G, \varphi_M).R_2 \subseteq R_1$.

DEFINITION 5.5. *Let* $\underline{B}_1 = \underline{B}(G_1, M_1, R_1)$ *and* $\underline{B}_2 = \underline{B}(G_2, M_2, R_2)$ *be concept lattices. Then* $\underline{B}_2$ *is a concept sublattice of* $\underline{B}_1$ *if* $\underline{B}_1$ *is homomorphic to* $\underline{B}_2$ *by some map h and there exist a context embedding* φ *such that the diagram in Figure 3 commutes (the function to concept lattice from context, described in Theorem 4.3, is denoted by the symbol* β*).*

The next theorem gives evidences that concept sublattices arise 'naturally'.

THEOREM 5.1. *Let* $\underline{B}_1 = \underline{B}(G_1, M_1, R_1)$ *and* $\underline{B}_2 = \underline{B}(G_2, M_2, R_2)$ *be concept lattices,* $\underline{B}_2$ *be compatible with* $\underline{B}_1$ *and let* $\underline{B}_1$ *be homomorphic to* $\underline{B}_2$ *by h. Then, an injective* φ *exists.*

Proof. Due to properties of h, we have that some $\underline{B}_s$, a sublattice of $\underline{B}_1$, isomorphic to $h.\underline{B}_2$, must exist. By Theorem 4.3, the context of this sublattice (G_s, M_s, R_s) can be determined. From Theorem 4.3 and Definition 4.3 it follows that any concept (C) of a concept lattice has more objects and (dually) fewer attributes than any concept smaller than C. By compatibility of $\underline{B}_2$ with $\underline{B}_1$ we have that $G_2 \subseteq G_s$ and $M_2 \subseteq M_s$, for some such $\underline{B}_s$. This implies that an injective φ can be defined. □

Theorem 5.1 only guarantees the existence of φ in the mathematical sense. In practice, context embedding may be subject to semantical conditions. Such conditions are beyond the scope of this paper.

In summary, the subsumption relation on concept lattices has 'two-levels'. First, the sublattice relation, in the sense of Definition 5.1, must be satisfied and secondly, an appropriate mapping of objects and attributes must exist.

The first of these conditions concerns the topology (or 'shape') of a concept lattice. This may be relevant in some practical applications, for example where the shape is the primary information and the elements may take their values from ranges. It is therefore important to know which changes of a context may leave the shape of the concept lattice unchanged.

Robustness of concept lattices is the subject of the two corollaries below. The first of them is based on the following observation. We can modify a concept (A, B) of $\underline{B}_1 = \underline{B}(G_1, M_1, R_1)$ by adding a new object, say $g \notin G_1$, to A. If the set of attributes of g is contained in B then the modified context (G_2, M_2, R_2), where $G_2 = G_1 \cup \{g\}$, $M_1 = M_2$ and $R_1 \subseteq R_2$, will be such that $\underline{B}_1 \cong \underline{B}_2$ and $(A \cup \{g\}, B)$ will be a concept of $\underline{B}_2$. The second corollary is similar to the first one, except that it allows elements of a context to be removed.

COROLLARY 5.1. *Extendability. Let* $\underline{B}_1 = \underline{B}(G_1, M_1, R_1)$ *and* $\underline{B}_2 = \underline{B}(G_2, M_2, R_2)$*, where* $G_2 = G_1 \cup G$, $M_2 = M_1$ *and* $R_2 \subseteq R_1 \cup G \times M_2$ *holds. If furthermore the condition* $G.R_2 \subseteq G_1.R_1$ *holds, then* $\underline{B}_1 \cong \underline{B}_2$*. (The dual statement holds, as well.)*

Proof. The condition, $G.R_2 \subseteq G_1.R_1$, ensures that for each element $g \in G$ there exists some element $g_1 \in G_1$, such that the set of attributes of g and that of g_1 are equivalent. From this and Theorem 4.3, part 2, the corollary follows immediately. □

COROLLARY 5.2. *Normalization. Let* $\underline{B}_1 = \underline{B}(G_1, M_1, R_1)$*. Then* $\underline{B}_2 = \underline{B}(G_2, M_2, R_2)$ *is called a normalized concept lattice of* $\underline{B}_1$*, if* $\underline{B}_1 \cong \underline{B}_2$, $G_1 \subseteq G_2$, $M_1 \subseteq M_2$, $R_1 \subseteq R_2$ *and furthermore* $R^r.G_2 = \emptyset$ *and* $R^l.M_2 = \emptyset$*.*

A context is normalized if the corresponding concept lattice is too.

The remaining part of this section is devoted to examples illustrating the usefulness of our definitions. The first of them serves also as a solution to the earlier subsumption problem. In this example it is assumed that an algorithm for deciding the concept sublattice relation, as in Definition 5.5, exists. Such an algorithm is described in Section 7. The second example contains applications of Corollaries 5.1 and 5.2.

EXAMPLE 2. Assume that the specifications and the concept lattices of the player (see Figure 1) and the card-game (see Figure 2) are given. It can be seen that Definition 5.5 can be applied to these concept lattices by the following assignments:

$$\varphi_G \in G_g \leftarrow G_p, \quad \varphi_M \in M_g \leftarrow M_p, \quad h \in \underline{B}_g \leftarrow \underline{B}_p,$$

$$\begin{array}{lll} \varphi_G(P_a) := P_0, & \varphi_M(l) := s_1, & h(C_0) := C_2, \\ \varphi_G(P_b) := P_1, & \varphi_M(r) := s_0. & h(C_2) := C_6, \\ \varphi_G(P_c) := P_3. & & h(C_3) := C_7, \\ & & h(C_1) := C_1. \end{array}$$

By this a player of the card game was found, namely, $\{C_1, C_2, C_6, C_7\}$. The other three players can easily be discovered. (Let us mention that there are four more sublattices of Figure 2, those having C_0 as bottom, for which an injective φ can not be defined.)

In summary, the concept lattice of the card game has four sublattices, isomorphic to the concept lattice of the player:

$$\begin{array}{ll} Y_0 = & \{C_1, C_2, C_6, C_7\}, \\ Y_1 = & \{C_1, C_3, C_6, C_8\}, \\ Y_2 = & \{C_1, C_4, C_8, C_9\}, \\ Y_3 = & \{C_1, C_5, C_7, C_9\}. \end{array}$$

Recall that the specification of the card game contains no mention of its players. Nevertheless they were found, due to the representation and Definition 5.5.

EXAMPLE 3. First, let us extend the specification of the player in the sense of Corollary 5.1, for example by adding the pair (P_d, l). One might think of such a pair as a simulation for 'noise'. If the concept lattice of the modified context is constructed, it can be seen that the shape of the concept lattice is the same as before.

Secondly, let us remove some pairs, e.g. (P_a, l) and (P_a, r), from the specification of the player. Again, this can be seen as a simulation for incomplete input due to noise. Corollary 5.2 shows that the shape of the concept lattice also remains unchanged in this case.

The last examples illustrates that the particular lattice representation is robust and to some extent fault-tolerant, as it can eliminate noise, for example in the first case, and repair incompleteness, for example, in the second case above. Fault tolerance has, of course, limitations, but an analysis of this is beyond the scope of this paper. It should be mentioned that context extension and normalization may change the meaning of a concept lattice.

6. REPEATED APPLICATION OF A GALOIS CONNECTION

This section shows how a Galois connection, the polars, can be repeatedly applied. The basic insight is that a concept lattice is itself a set with a partial ordering on it. There are various ways of applying the results developed so far, but we will mostly concentrate on one of them which provides an unexpected result.

Looking closer at the definition of a Galois connection (Definitions 3.1, 3.4 and 3.5) it is noticeable that the parameters of the pair of functions (R^l, R^r) are G, M and, perhaps most importantly, the relation R.

The intention with 'repeated application' is to apply the Galois connection (R^l, R^r) to a concept lattice. By this, the choice of the sets G and M are slightly restricted. Actually, only two cases will be considered. In the first one, the set of elements of a concept lattice to define G and M were chosen and in the second one, the set of subsets of those elements.

Assume $\underline{B} = (B, \leq)$ is a concept lattice ($\leq$ denotes the usual ordering of concepts introduced in Definition 4). We define R_c, a relation on concepts as follows.

DEFINITION 6.1. *Let $G_c = B$ and $M_c = B$. Then for all* $\mathsf{a}, \mathsf{b} \in B$, $\mathsf{a} R_c \mathsf{b}$ *iff* a *covers* b, *that is,* $\mathsf{b} < \mathsf{a}$ *and there is no element* x *in B, such that* $\mathsf{b} < \mathsf{x} < \mathsf{a}$.

The context (G_c, M_c, R_c) is theoretically interesting, but it was not found directly relevant in practice.

A generalization of R_c to a relation on sublattices offers better prospects, let alone the sublattices are also attractive for other reasons as stated by the next theorem.

THEOREM 6.1. *([13]) For a lattice L we denote the set of sublattices by $\mathcal{P}_s.L$. Then $\mathcal{P}_s.L \cup \{\emptyset\}$ is a complete lattice.*

The set-valued function $\mathcal{P}_s$ exists as only finite sets are considered. The special element $\{\emptyset\}$ is added for reasons of completeness of the lattice.

Again, assume $\underline{B} = (B, \leq)$ is a concept lattice and let $(S, \subseteq) := \mathcal{P}_s.\underline{B} \cup \{\emptyset\}$ denote the set of sublattices of $\underline{B}$. We define R_s, a relation on sets of concepts (sublattices), by a generalization of a cover from singletons to sets.

DEFINITION 6.2. *Let $G_s = S$ and $M_s = S$. Then for* $\mathsf{a}, \mathsf{b} \in S$ *(*$\mathsf{a} \not\subseteq \mathsf{b}$ *and* $\mathsf{b} \not\subseteq \mathsf{a}$*)* $\mathsf{a} R_s \mathsf{b}$ *iff* $\exists \mathsf{x} \in \mathsf{a}$ *and* $\exists \mathsf{y} \in \mathsf{b}$ *(*x *and* y *not containing the empty set) such that* $\mathsf{x} R_c \mathsf{y}$ *or* $\mathsf{y} R_c \mathsf{x}$ *holds.*

It has been shown that a concept lattice can represent a specification. By a repeated application of a Galois connection a concept lattice (of some larger specification) can be decomposed to smaller concept lattices and thereby a new conceptualization of the former one can be found.

DEFINITION 6.3. *A decomposition of a set B is a collection of subsets, $\{B_i \mid i \in I\}$, such that for $i \in I$ the B_is are all different and their set union is B. It is also required that $B_i \not\subseteq B_j$ for all $i, j \in I$ $(i \neq j)$.*

DEFINITION 6.4. *Let $\underline{B} = (B, \leq)$ be a concept lattice and $\{\underline{B}_j \mid j \in J\}$ a collection of concept lattices. A decomposition of a concept lattice $\underline{B}$ is a decomposition of B, $\{B_i \mid i \in I\}$, such that, for all $i \in I$: $\underline{B}_i = (B_i, \leq)$ is a concept sublattice of $\underline{B}$ and $\underline{B}_i \cong \underline{B}_j$ for some $j \in J$.*

EXAMPLE 4. In Example 2 it was shown that the lattice of the card-game contains the sublattices Y_0–Y_3. Let us take them as our objects, and as attributes, the sublattices related to them by Definition 6.2.

In order to make the formulae more readable, I will introduce the prefix 'with' for a named subset when used as an attribute. Eventually, by Definition 6.2, we obtain the following relation between the sublattices Y_0–Y_3: $R_s = \{(Y_0, \text{with } Y_1), (Y_0, \text{with } Y_3), (Y_1, \text{with } Y_0), (Y_1, \text{with } Y_2), (Y_2, \text{with } Y_1), (Y_2, \text{with } Y_3), (Y_3, \text{with } Y_0), (Y_3, \text{with } Y_2)\}$.

By applying the Galois connection described in Section 4 again, we obtain the concepts:

$$\begin{aligned} C_0 &= (\{Y_0, Y_1, Y_2, Y_3\}, \emptyset), \\ C_1 &= (\{Y_0, Y_2\}, \{\text{with } Y_1, \text{with } Y_3\}), \\ C_2 &= (\{Y_1, Y_3\}, \{\text{with } Y_0, \text{with } Y_2\}), \\ C_3 &= (\emptyset, \{\text{with } Y_0, \ldots, \text{with } Y_3\}). \end{aligned}$$

Topologically, this lattice is isomorphic to the one of the player (see Figure 1), but as a concept lattice it now has a completely different meaning: the card-game conceptualized as pairs of players. Indeed, it is observed that C_1 and C_2 (those concepts that do not contain the empty set) have a pair of players as object resp. attribute sets.

7. AN ALGORITHM

The most important part of a lattice subsumption algorithm is that of lattice embedding. An implementation of φ, a context embedding, may not in general be difficult.

DEFINITION 7.1. *For a lattice $L = (V, \leq)$ the corresponding graph $G = (V, E)$ is such that for $a, b \in V$:*

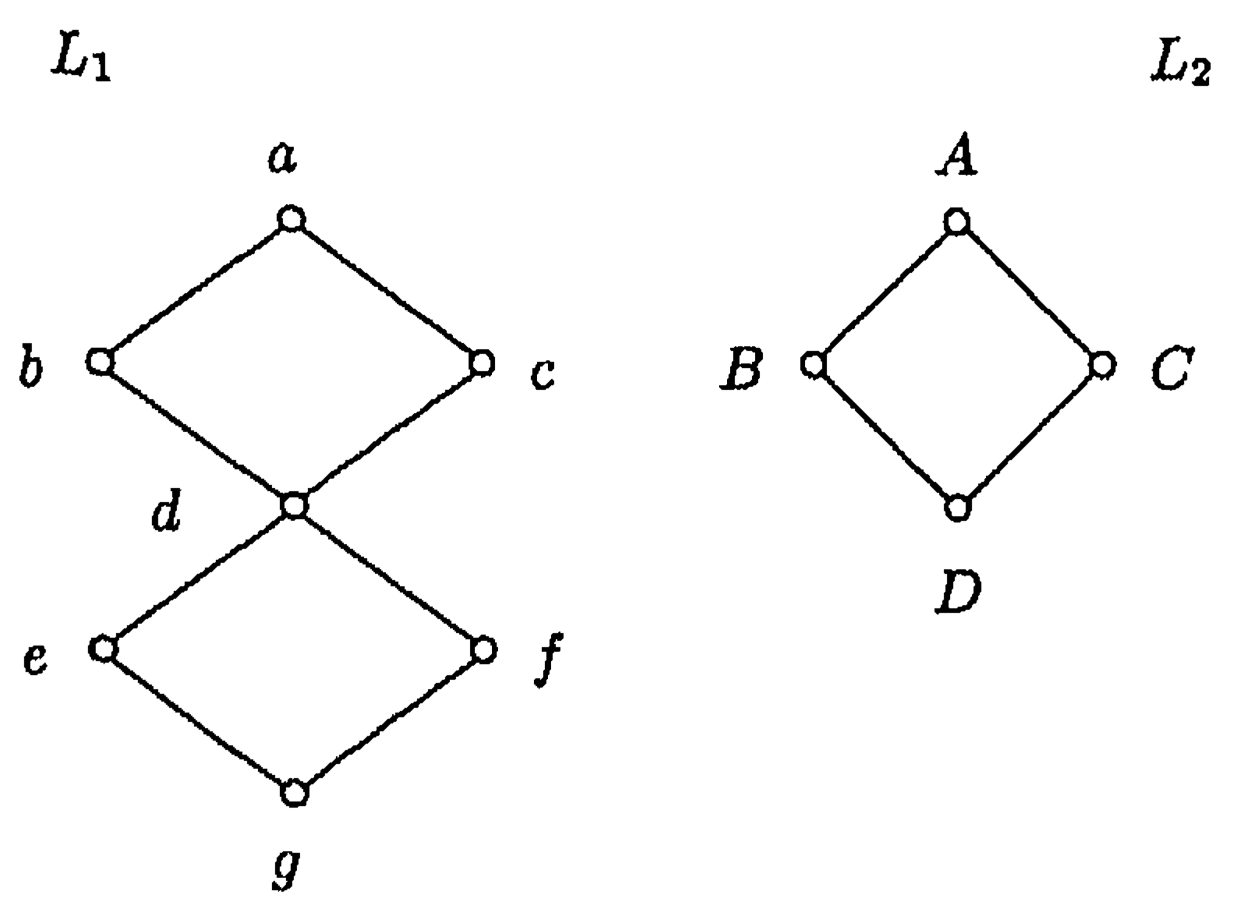

FIGURE 4. Sample lattices.

$(a, b) \in E$ if a covers b. The set of parents of a node v is defined as $out(v) = \{w \mid (v, w) \in E\}$.

DEFINITION 7.2. *Let L be a lattice, the elements of L ordered by $\sqsubseteq$ and $x \in L$. We define $\Delta(x)$, the up-set of x, as $\Delta(x) = \{y \in L \mid x \sqsubseteq y\}$; and $\nabla(x)$, the down-set of x, as $\nabla(x) = \{y \in L \mid y \sqsubseteq x\}$.*

We describe an algorithm for lattice embedding in [14]. This algorithm (see Figure 5) determines whether L_1 contains a sublattice isomorphic to L_2 for the two lattices L_1 and L_2.

The lattices are represented as graphs, and the nodes of L_2 are partitioned according to their depth (the depth of a node is the length of the shortest path from the root to that node).

The algorithm traverses the graph of L_1 top-down and the partitions of L_2 in depth order. In each step, nodes of a partition of L_2 are mapped to those nodes of L_1 that are 'below' the nodes already involved in a map of some earlier partition. Having found a map, the algorithm checks whether it is order preserving.

EXAMPLE 5. [14] Consider the lattices L_1 and L_2 of Figure 4. Partitioning of V_2 by depth yields $\{(A), (B, C), (D)\}$. Below we show the stepwise computation of the match $M = \{(A, a), (B, c), (C, b), (D, e)\}$.

$(k = 1)$ $m = \{(A, a)\}$
$(k = 2)$ $M_2 = \{a\}$, $\nabla(M_2) = \{b, c, d, e, f, g\}$;
$m = \{(B, c), (C, b)\}$
This is a well-connected match because
$(C, A) \in E_2 : (b, a) \in E_1^*$ and
$(B, A) \in E_2 : (c, a) \in E_1^*$
$(k = 3)$ $M_3 = \{a, b, c\}$, $\nabla(M_3) = \{d, e, f, g\}$;
$m = \{(D, e)\}$
This is a well-connected match because
$(D, B) \in E_2 : (e, c) \in E_1^*$ and
$(D, C) \in E_2 : (e, b) \in E_1^*$
$(k = 4)$ $M_4 = \{a, b, c, e\}$

At this point all elements of V_2 have an image in L_1. The matching has to be verified to check if it is a sublattice of L_1. As $d = c \sqcap b = b \sqcap c$ and $d \notin M_4$ we have to reject this matching for not being a sublattice. If, however, we had (D, d) instead of (D, e) the matching would have been accepted.

Input $L_1 = (V_1, E_1^*)$, where E_1^* is the reflexive, transitive closure of E_1; $L_2 = (V_2', E_2)$ with V_2' partitioned in classes $C_1, .., C_n$, where $C_i = \{v \in V_p \mid depth(v) = i\}$, for $1 \leq i \leq n$.
Output All sublattices of L_1 isomorphic to L_2, represented by sets of tuples M, where a tuple $(x, y) \in M$ denotes the matching of $x \in L_2$ with $y \in L_1$.

procedure *match (k, M):*
begin
 if $k = 1$ **then**
  for $v_1 \in V_1$ **do**
   $m := \{(\top_2, v_1)\}$;
   match$(k + 1, M \cup m)$ **od**
 elseif $1 < k \leq n$ **then**
  $M_k := \{b \mid (a, b) \in M\}$;
  $\nabla(M_k) := \{a \mid \exists_{c \in M_k} : (a, c) \in E_1^* \text{ and } a \notin M_k\}$;
  for *every matching m between* C_k *and*
   a subset X of $\nabla(M_k)$ *with* $|X| = |C_k|$ **do**
   if $\forall_{a \in C_k} \forall_{b \in out(a)}$
    $[(a, x) \in m \text{ and } (b, y) \in M \Rightarrow (x, y) \in E_1^*]$
    then *match*$(k + 1, M \cup m)$
   endif od
 else
  {*k corresponds to* $n + 1$}
  lattice condition := **true**;
  for $(a_1, b_1) \in M$, $(a_2, b_2) \in M$ **do**
   if $\neg\exists a$: $(a, b_1 \sqcup b_2) \in M$ or $(a, b_1 \sqcap b_2) \in M$
   then *lattice condition* := **false endif od**
  if *lattice condition* **then** *output(M)* **endif**
 endif
end
begin
 match$(1, \emptyset)$
end

FIGURE 5. A top-down algorithm for lattice embedding.

The complexity of lattice embedding is exponential [14]. The algorithm can only be optimized by considering the irreducible elements of L_2. This optimization, however, does not change the worst case complexity, as the number of irreducible elements of a lattice $L = (V, \leq)$ is in the order of $|V|$.

8. DISCUSSION

Why use concept lattices and not some other formalism, like Prolog? The question is proper, so how might the running example look in that language? The following specification is not complete, we have only included the essential parts. We denote a person (`Xi`) seated at some side of a table (`Li`) by the pair `[Xi, Li]`, for $0 \leq i \leq 3$; variable `Pair_List` denotes a list of such pairs.

```
player([X0,L0],[X0,L1],[X1,L0],[X2,L1]):-
  X0 ≠ X1, X0 ≠ X2, X1 ≠ X2,
  L0 ≠ L1, L0 ≠ L2, L1 ≠ L2.

card_game([X0,L0],[X0,L1],[X1,L0],[X1,L2],
        [X2,L2],[X2,L3],[X3,L3],[X3,L0]):-
  X0 ≠ X1, X0 ≠ X2, X1 ≠ X2, X3 ≠ X2,
  X3 ≠ X1, X3 ≠ X0,
  L0 ≠ L1, L0 ≠ L2, L1 ≠ L2, L3 ≠ L2,
  L3 ≠ L1, L3 ≠ L0.

player of card_game (Pair_List):-
  select pairs(Pair_List,[X0,L0],
    [X0,L1],[X1,L0],[X2,L1]),
  player([X0,L0],[X0,L1],[X1,L0],[X2,L1]).

select pairs (Pair_List, [X0,L0],
  [X0,L1], [X1,L0], [X2,L1]):-
  choose arbitrary pairs from Pair_List and
  determine thereby the values of X0, X1, X2
    and L0, L1, L2.
```

The complicated predicate `select pairs` hides most of the computational work that is needed to identify some person–side pairs as a player. The other hidden tool, backtracking, is used to find all occurrences of the player.

So far the implementation issues of concept lattices have not been considered. Here, we only mention that Prolog implementations (those we are aware of) are unable to benefit from the lattice structure of the domain of the variables. So, Prolog might be less efficient in our case.

This is, however, just a minor point. A more important feature of our Prolog example is that the clauses are complicated (besides the predicate `select pair` we also mention the right-hand sides of the clauses `player` and `card_game` which specify the uniqueness of their points and lines). Complicated specifications are difficult to obtain complete and correct. Concept lattices allow more systematic specifications. This is, amongst others, why this approach may have advantages over other formalisms.

9. SUMMARY

Concept lattices can be used to represent specifications (knowledge) and determine a subsumption relation on those specifications in a uniform and systematic way. Repeated application of concept lattice construction may provide us with new conceptualizations of the data. I applied my theory to a pair of specifications and made a comparison with another approach that is based on Prolog.

My current research in the area of concept lattices includes: (i) applying concept sublattices in transformational program specification, and (ii) natural language processing. The former focuses on the question: how top-down and bottom-up specifications are actually combined. The latter concentrates on two problems: how natural language can be modelled by using concept lattices, and how such a model can help to bring the fields of conceptual lattices and conceptual graphs closer.

ACKNOWLEDGEMENTS

The author would like to thank Hans Meijer for his valuable suggestions and József Farkas for his various contributions to this work.

REFERENCES

[1] Wille, R. (1983) Subdirect decomposition of concept lattices. *Algebra Universalis*, **17**, 275–287.

[2] Carpineto, C. and Romano, G. (1993) GALOIS: An order-theoretic approach to conceptual clustering. In *Proc. of the 10th Int. Conf. on Machine Learning*, Amherst, MA, pp. 33–40.

[3] Godin, R., Missaoui, R. and Alaoui, H. (1995) Incremental concept formation algorithms based on Galois (concept) lattices. *Computat. Intelligence*, **11**, 246–267.

[4] Godin, R. and Missaoui, R. (1994) An incremental concept formation approach for learning from databases. *Theor. Comput. Sci.*, **133**, 387–419.

[5] Aart, C., Backhouse, R., Hoogendijk, P., Voermans, E. and van der Woude, J. (1992) *A relational theory of datatypes*. In preparation: available by ftp from `ftp.win.tue.nl`.

[6] Wille, R. (1982) Restructuring lattice theory: an approach based on hierarchies of concepts. In Rival, I. (ed.), *Ordered Sets*, pp. 445–470. D. Reidel, Dordrecht.

[7] Birkhoff, G. (1967) *Lattice theory*, 3rd edn. American Mathematical Society, Providence, RI.

[8] Ganter, B. and Wille, R. (1989) Conceptual scaling. In Roberts, F. (ed.), *Applications of Combinatorics and Graph Theory to the Biological and Social Sciences*, pp. 139–167, Springer-Verlag, Berlin.

[9] Wille, R. (1989) Knowledge acquisition by methods of formal concept analysis. In Diday, E. (ed.), *Proc. of the Conf. on Data Analysis, Learning Symbolic and Numeric Knowledge*, pp. 365–380. Nova Science Publishers, Inc, New York.

[10] Sarbo, J. and Farkas, J. (1994) Concept sublattices. In *Proc. of the European Conference on Machine Learning (ECML'94)*, April 6–8 Catania, Italy. *Lecture Notes in Artificial Intelligence*, **784**, Springer-Verlag, Berlin.

[11] Grice, H. P. (1975) Logic and conversation. *Syntax Semiotics: Speech Acts*, **3**, 41–58.

[12] Sarbo, J. J. and Farkas, J. I. (1995) Knowledge representation and acquisition by concept lattices. In Markovitch, S. (ed.), *Proc. of the 11th Israeli Symposium on Artificial Intelligence (ISAI'95)*, January 17–18. Hebrew University of Jerusalem, Israel.

[13] Davey, B. A. and Priestley, H. A. (1990) *Introduction to Lattices and Order*. Cambridge University Press, UK.

[14] Sarbo, J. J. (1996) Lattice embedding. In Eklund, P. W., Ellis, G. and Mann, G. (eds), *Conceptual Structures: Knowledge Representation as Interlingua (ICCS'96)*, **1115**, *Springer-Verlag*, 293–307.