The following full text is a publisher's version.

For additional information about this publication click this link.
http://hdl.handle.net/2066/28107

Please be advised that this information was generated on 2018-11-21 and may be subject to change.
Building Sub-Knowledge Bases
Using Concept Lattices

J. J. SARBON
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 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.

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)
was called a context. The lattice arises from that context by
applying a Galois connection between the power sets of
objects and the set of attributes $M$. This Galois connection is a particular one, called the
classic [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 $A' \subseteq G$ the answer is a concept $(A, B)$ of the concept lattice, such that $A' \subseteq A$ 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).

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 $\mathcal{A} = (A, \subseteq_A)$ and $\mathcal{B} = (B, \subseteq_B)$ to be posets (partially ordered sets). When this will not lead to confusion the substrings 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 $A \subseteq A$ the lifted function $F$ is defined by $F.A = \{a : a \in A : F.a\}$.

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

$$\forall(x, y : x \in B \land y \in A : F_1.x \subseteq y = x \subseteq F_2.y).$$

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 $A$ or $B$ are complete lattices then $F_1, B$ and $F_2, A$ are isomorphic complete lattices.

Later Theorem 3.1 will be applied to the particular case $\mathcal{A} = (P.A, \subseteq)$ and $\mathcal{B} = (P.B, \supseteq)$ where $A$ and $B$ denote a set of objects and a set of attributes, respectively, and $P$ 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 $P.B$ from $A$ by defining for every $a \in A : a.R = \{b : b \in B : a.Rb\}$.

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

The functions $a.R$ and $R.b$ can be lifted to functions to $P.B$ from $P.A$, and to $P.A$ from $P.B$, 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:

$$[B]^L = \cap \{ b : b \in B : R.b \}.$$  

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

THEOREM 3.2. The pair of polars $(R^A, R^L)$ 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) $A' = \{ m \in M : \forall g \in A \ gRm \}$ for $A \subseteq G$,
b) $B' = \{ g \in G : \forall m \in M \ gRm \}$ 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 $B' = [B]^L$. 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:

$$[G]^R \supseteq M \iff G \subseteq [R]^M.$$  

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 $M \in \mathcal{P}.M$, $[M]^L$ contains those objects that are common for all attributes in $M$.

DEFINITION 4.2. Concept. A concept is a pair $(G, M)$ with $G = [M]^R = R^L.M$ and $M = [G]^R = R^G.$

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

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

$$(G_1, M_1) \sqsubseteq (G_2, M_2) \iff G_1 \subseteq G_2.$$  

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

Proof: $G_1 \subseteq G_2$

$\Rightarrow \{ \text{concept} \}$

$\Rightarrow \{ \text{Theorem 4.1} \}$

$\Rightarrow \{ G_1 \} R \supseteq M_2$

$\Rightarrow \{ \text{concept} \}$

$M_1 \supseteq M_2$. □

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 $R_p = \{ (P_a, l), (P_b, r), (P_b, l), (P_c, r) \}$.

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 $[A]^R$ of $M_p$ must be computed. Let us begin with the subset $A = \{ P_a \}$. Since $A$ is a singleton set, we find that $[A]^R = \{ l, r \}$ (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: $R_p(\{ l, r \}) = R_p(\{ l \}) \cap R_p(\{ r \}) = \{ P_a \} \cap \{ P_a \} = \{ P_a \}$. We may conclude that $(\{ P_a \}, \{ l, r \})$ is a concept.

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

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

$C_0 = (\{ P_a \}, \{ l, r \}),$
$C_1 = (\{ P_a, P_b, P_c \}, \{ \} ),$
$C_2 = (\{ P_a, P_b \}, \{ l \}),$
$C_3 = (\{ P_a, P_c \}, \{ r \}).$
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 sublatice 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 \( a \cup b \in X \) and \( a \cap b \in X \).

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 \( L_1 \) to \( L_2 \). Then \( L_1 \) is homomorphic to \( L_2 \) by \( h \). If \( h \) is bijective, it is called an isomorphism.

DEFINITION 5.3. Let \( B_1 = \mathcal{B}(G_1, M_1, R_1) \) and \( B_2 = \mathcal{B}(G_2, M_2, R_2) \) be concept lattices. Let us denote their bottom and top elements, for \( i = 1, 2 \), by \( \bot_i = (G_i^\downarrow, M_i^\uparrow) \) and \( \top_i = (G_i^\uparrow, M_i^\downarrow) \). We say \( B_2 \) is compatible with \( B_1 \) if \( G_2^\downarrow \subseteq G_1^\downarrow \) and \( M_2^\uparrow \subseteq M_1^\uparrow \).

DEFINITION 5.4. A mapping between contexts, \( \varphi \in (G_1, M_1, R_1) \dashv (G_2, M_2, R_2) \), is called a context
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 \( \mathcal{B}_1 = \mathcal{B}(G_1, M_1, R_1) \) and \( \mathcal{B}_2 = \mathcal{B}(G_2, M_2, R_2) \) be concept lattices. Then \( \mathcal{B}_2 \) is a concept sublattice of \( \mathcal{B}_1 \), if \( \mathcal{B}_1 \) is isomorphic to \( \mathcal{B}_2 \) by some map \( h \) and there exist a context embedding \( \varphi \) 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 \( \beta \)).

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

**Theorem 5.1.** Let \( \mathcal{B}_1 = \mathcal{B}(G_1, M_1, R_1) \) and \( \mathcal{B}_2 = \mathcal{B}(G_2, M_2, R_2) \) be concept lattices, \( \mathcal{B}_2 \) be compatible with \( \mathcal{B}_1 \), and let \( \mathcal{B}_1 \) be isomorphic to \( \mathcal{B}_2 \) by \( h \). Then, an injective \( \varphi \) exists.

**Proof.** Due to properties of \( h \), we have that some \( \mathcal{B}_3 \), a sublattice of \( \mathcal{B}_1 \), isomorphic to \( \mathcal{B}_2 \), must exist. By Theorem 4.3, the context of this sublattice \( (G_3, M_3, R_3) \) 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 \( \mathcal{B}_2 \) with \( \mathcal{B}_1 \), we have that \( G_2 \subseteq G_1 \) and \( M_2 \subseteq M_1 \), for some such \( \mathcal{B}_3 \). This implies that an injective \( \varphi \) can be defined.

Theorem 5.1 only guarantees the existence of \( \varphi \) 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 \( \mathcal{B}_1 = \mathcal{B}(G_1, M_1, R_1) \) by adding a new object, say \( g \not\in G_1 \), to \( A \). If the set of attributes of \( g \) is contained in \( B \) then the modified context \( (G_2, M_2, R_2) \), where \( G_2 = G_1 \cup \{g\}, M_2 = M_1 \) and \( R_1 \subseteq R_2 \), will be such that \( \mathcal{B}_1 \cong \mathcal{B}_2 \) and \( (A \cup \{g\}, B) \) will be a concept of \( \mathcal{B}_2 \). 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 \( \mathcal{B}_1 = \mathcal{B}(G_1, M_1, R_1) \) and \( \mathcal{B}_2 = \mathcal{B}(G_2, M_2, R_2) \), where \( G_2 = G_1 \cup G, M_2 = M_1 \) and \( R_2 \subseteq R_1 \cup G \times M_1 \) holds. If furthermore the condition \( G.R_2 \subseteq G_1.R_1 \) holds, then \( \mathcal{B}_1 \cong \mathcal{B}_2 \). (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 \( \mathcal{B}_1 = \mathcal{B}(G_1, M_1, R_1) \). Then \( \mathcal{B}_2 = \mathcal{B}(G_2, M_2, R_2) \) is called a normalized concept lattice of \( \mathcal{B}_1 \), if \( \mathcal{B}_1 \cong \mathcal{B}_2 \), \( G_1 \subseteq G_2, M_1 \subseteq M_2, R_1 \subseteq R_2 \) and furthermore \( R^\perp \). \( G_2 = \emptyset \) and \( R^\perp \). \( M_2 = \emptyset \).

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 context 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. U can be seen that by compatibility of \( G.R \) and \( M \), we have that some \( \mathcal{B}_1 \), isomorphic to \( \mathcal{B}_2 \), must exist. By 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 \( \mathcal{B}_2 \) with \( \mathcal{B}_1 \), we have that \( G_2 \subseteq G_1 \) and \( M_2 \subseteq M_1 \), for some such \( \mathcal{B}_3 \). This implies that an injective \( \varphi \) can be defined.

Theorem 5.1 only guarantees the existence of \( \varphi \) 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 \( \mathcal{B}_1 = \mathcal{B}(G_1, M_1, R_1) \) by adding a new object, say \( g \not\in G_1 \), to \( A \). If
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 \((P_a, l)\). 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 \((R^1, R^r)\) are \(G, M\) and \(R\), and perhaps most importantly, the relation \(R\).

The intention with 'repeated application' is to apply the Galois connection \(R^1, R^r\) to a concept lattice. By this, the choice of the sets \(G, M\) and \(R\) are slightly restricted. Actually, only two cases will be considered. In the first one, the set of elements of a concept lattice to define \(G, M\) and \(R\) is chosen and in the second, the set of subsets of those elements.

Assume \(\mathcal{B} = (B, \leq)\) 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 \(a, b \in B\), \(a R_c b\) if \(a\) covers \(b\), that is, \(b < a\) and there is no element \(x \in B\) such that \(b < x < a\).

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 an unexpected result.

**Theorem 6.1.** \((135)\) For a lattice \(L\) we denote the set of sublattices by \(\mathcal{P}_L\). Then \(\mathcal{P}_L\cup\{\emptyset\}\) is a complete lattice.

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

Again, assume \(\mathcal{B} = (B, \leq)\) is a concept lattice and let \((S, \subseteq)\) be a relation on sets of sublattices of \(\mathcal{B}\). We define \(R_s\), a relation on sets of sublattices, by a generalization of a cover from singletons to sets.

**Definition 6.2.** Let \(G_s = S\) and \(M_s = S\). Then for \(a, b \in S\) \((a \not\subset b\) and \(b \not\subset a\) \(a R_s b\) if \(\exists x \in a\) and \(\exists y \in b\) (\(x\) and \(y\) not containing the empty set) such that \(x R_y\) or \(y R_x\) 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 \(B_i\)s are all different and their set union is \(B\). It is also required that \(B_i \subsetneq B_j\) for all \(i, j \in I\) \((i \neq j)\).

**Definition 6.4.** Let \(\mathcal{B} = (B, \leq)\) be a concept lattice and \(\{B_j \mid j \in J\}\) a collection of concept lattices. A decomposition of a concept lattice \(\mathcal{B}\) is a decomposition of \(B\), \(\{B_i \mid i \in I\}\), such that, for all \(i \in I\) : \(\mathcal{B}_i = (B_i, \subseteq)\) is a concept sublattice of \(\mathcal{B}\) and \(\mathcal{B}_i \cong \mathcal{B}_j\) 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, with Y_1), (Y_0, with Y_2), (Y_1, with Y_0), (Y_2, with Y_1), (Y_1, with Y_2), (Y_3, with Y_0), (Y_2, with Y_3), (Y_3, with Y_2), (Y_0, with Y_2), (Y_2, with Y_0)\}.

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

\[\begin{align*}
C_0 &= \{(Y_0, Y_1, Y_2, Y_3), \emptyset\}, \\
C_1 &= \{(Y_0, Y_2), \{with Y_1, with Y_3\}\}, \\
C_2 &= \{(Y_1, Y_2), \{with Y_0, with Y_2\}\}, \\
C_3 &= \{\emptyset, \{with Y_0, \ldots, with Y_3\}\}.
\end{align*}\]

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 \(\varphi\), 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 \(a, b \in V:\)
Let $L$ be a lattice, the elements of $L$ ordered by $\leq$ and $x \in L$. We define $\Delta(x)$, the up-set of $x$, as $\Delta(x) = \{ y \in L \mid x \leq y \}$; and $\nabla(x)$, the down-set of $x$, as $\nabla(x) = \{ y \in L \mid y \leq x \}$.

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 $M = \{(A, a), (B, c), (C, b), (D, e)\}$.

$$(k = 1) \quad m = \{(A, a)\}$$

$$(k = 2) \quad M_2 = \{(a), \nabla(M_2) = \{b, c, d, e, f, g\};
\quad m = \{(B, c), (C, b)\}$$

This is a well-connected match because $(C, A) \in E_2 : (b, a) \in E_1^*$ and $(B, A) \in E_2 : (c, a) \in E_1^*$.

$$(k = 3) \quad M_3 = \{(a, b, c), \nabla(M_3) = \{d, e, f, g\};
\quad m = \{(D, e)\}$$

This is a well-connected match because $(D, B) \in E_2 : (e, c) \in E_1^*$ and $(D, C) \in E_2 : (e, b) \in E_1^*$.

$$(k = 4) \quad M_4 = \{(a, b, c, e)\}$$

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 $d = c \cap b = b \cap c$ and $d \notin M_4$ we have to reject this matching for not being a sublattice. If, however, we have $(D, d)$ instead of $(D, e)$ the matching would have been accepted.

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 $(X_i)$ seated at some side of a table $(L_i)$ by the pair $[X_i, L_i]$, 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 in-  
cludes: (i) applying concept subtlettes 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

Algebra Universalis, 17, 275–287.
theoretic approach to conceptual clustering. In Proc. of the  
10th Int. Conf. on Machine Learning, Amherst, MA, pp. 33–  
40.
concept formation algorithms based on Galois (concept)  
formation approach for learning from databases. Theor.  
preparation: available by ftp from ftp .win. tue. nl.
based on hierarchies of concepts. In Rival, I. (ed.), Ordered  
Mathematical Society, Providence, RI.
Roberts, F. (ed.), Applications of Combinatorics and Graph  
Theory to the Biological and Social Sciences, pp. 139–167,  
Springer-Verlag, Berlin.
concept analysis. In Diday, E. (ed.), Proc. of the Conf. on  
Data Analysis, Learning Symbolic and Numeric Knowledge,  
Proc. of the European Conference on Machine Learning  
(ECML’94), April 6–8 Catania, Italy. Lecture Notes in  
Artificial Intelligence, 784, Springer-Verlag, Berlin.
and acquisition by concept lattices. In Markovitch, S. (ed.),  
Proc. of the 11th Israeli Symposium on Artificial Intelligence  
(ISA’95), January 17–18. Hebrew University of Jerusalem,  
Israel.
Lattices and Order. Cambridge University Press, UK.
W., Ellis, G. and Mann, G. (eds), Conceptual Structures:  
Knowledge Representation as Interlingua (ICCS’96), 1115,  
Springer-Verlag, 293–307.