TITLE PAGE

| | |
|---|---|
| Title: | BUILDING SUB-KNOWLEDGE BASES USING CONCEPT LATTICES |


| | |
|---|---|
| Author: | J.J. Sarbo |
| Address: | University of Nijmegen |
| | Toernooiveld 1, 6525 ED Nijmegen |
| | The Netherlands |
| Tel.nr.: | 31-024-3653049 |
| Fax nr.: | 31-024-3553450 |
| e-mail: | janos@cs.kun.nl |

# Building sub-knowledge bases using concept lattices

J.J. Sarbo
*Computing Science Institute, University of Nijmegen,*
*Toernooiveld 1, 6525 ED Nijmegen, The Netherlands*
*E-mail: janos@cs.kun.nl*

## Abstract

A theory of concept (Galois) lattices was first introduced by R. Wille ([1]). An extension of his work to simple structures called concept sublattices was given in [2].

In this paper we show that concept sublattices can be applied for (i) determining subsumption of specifications, and (ii) decomposing specifications in terms of other ones. We show that the latter application of the theory may provide us with new conceptualizations of a specification.

## 1 Introduction

We consider the following problem: suppose we want to specify notions, like player, card-game, bridge etc., and describe each of them in terms of a set of primitives, e.g. person, seating etc. So, for example, a kind of player is specified as a person who plays a game against his neighbors (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 our specification and determine the relationship between the specified notions, e.g. whether the card-game has players, then we find ourselves in an inconvenient situation. Though all necessary information is present in the specification, it is given in terms of persons, topology etc., 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 etc. 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.

In this paper we show 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. We also show that by determining such a subsumption relation on specifications we may find decompositions, or even new conceptualizations (*sub-knowledge base*) of a specification.

Decomposition of concept lattices is found in [3]. We should emphasize that the subdirect decomposition of [3] differs from the one described in this paper, as we investigate decomposition of concept lattices in terms of a set of given ones and we allow embedding of contexts (cf. section 5).

Batch and incremental algorithms for building concept lattices are found in [4] and [5]. Automatic discovery of implication rules from data using concept lattices is described by Godin and Missaoui ([6]). Their approach is close to ours one, 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. In section 3 we recall basic definitions and theorems of lattice theory. The terminology and much of the presentation is borrowed, with appropriate adjustments, from [7]. Section 4 recapitulates the basic notions and a fundamental theorem on concept lattices. Section 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 we compare our approach with one, using Prolog.

## 2 Concept lattices

Concept lattices were first defined by R. Wille ([1]) 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* ([8]).

Various applications of concept lattices have been reported in the literature ([9, 5, 6, 10]). Typically, such an application assumes a subset of $G$ (resp. $M$) to be given as *input* for which a corresponding subset of $M$ (resp. $G$) 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 e.g. given input $\mathsf{A}' \subseteq G$ the answer is a concept $(\mathsf{A}, \mathsf{B})$ of the concept lattice, such that $\mathsf{A}' \subseteq \mathsf{A}$ and $\mathsf{A}$ is "smallest". Then $\mathsf{B}$ is the corresponding subset of $M$.

In this sense a concept lattice is an appropriate representation for what is referred to in 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 [2, 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 be represented by means of a Galois connection, as well. Eventually this generalization allows the input to be a concept lattice.

For such an input, similarly to our previous example, the answer is the "smallest" sublattice of the concept lattice (knowledge base) that subsumes it. In sum, 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 base"). We will show that in that 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 get an abstract conceptualization of a specification.

An interesting question one may raise is whether the concept lattice construction could be repeatedly applied, now to the structure resulted by such a decomposition. The answer is, somewhat unexpectedly, affirmative and in this paper we will show that various ways for a recursive application are possible.

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 be arbitrarily 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 (cf. section 6).

# 3 Basic definitions

In this section we recall some useful definitions and theorems about Galois connections. As indicated above, we make extensive use of [7] and prove "borrowed" theorems only when it does not prove them. Another source of savings is due to duality: the dual of a theorem may be stated without proving it.

In what follows we take $A$ and $B$ to be sets, $\mathsf{A}$ and $\mathsf{B}$ 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 we drop the subscripts from the orderings. We 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

[1] In a nutshell, cooperative communication ([11]) is characterized by four maxims of which this representation may help to realize the first two: (1) (*maxim of quantity*) make your contribution as informative as required, and not more than that, (2) (*maxim of quality*) try to make your contribution one that is true, (3) (*maxim of relation*) make your contribution relevant w.r.t. the previous one, (4) (*maxim of manner*) be perspicuous.

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 3.2** 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 we will apply theorem 3.2 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 we will consider are finite.

We mentioned that polars ([8]) play a central rôle in concept lattices. We will define polars stepwise, starting from a set of more simple functions.

**Definition 3.3** For 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 3.4** For 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 3.5** For every $\mathsf{A} \in \mathcal{P}.A$ we define the *right polar*:

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

**Definition 3.6** For every $\mathsf{B} \in \mathcal{P}.B$ we define the *left polar*:

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

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

**Theorem 3.7** The pair of polars $(R^r, R^l)$ is a Galois connection.

A proof of this theorem is found in [7]. Birkhoff ([8]) applies this theorem as a definition for Galois connections.

# 4 Formal concepts

We mentioned that 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 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 [1].

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

$$\langle \mathsf{G} \rangle R \supseteq \mathsf{M} \ \equiv\ \mathsf{G} \subseteq R\langle \mathsf{M} \rangle \ .$$

**Proof** A consequence of definition 3.1 by the substitution $\langle \_ \rangle R$ for $F_2$ and $R\langle \_ \rangle$ for $F_1$ .

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

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

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

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

$$(\mathsf{G}_1, \mathsf{M}_1) \leq (\mathsf{G}_2, \mathsf{M}_2) \equiv \mathsf{G}_1 \subseteq \mathsf{G}_2 \ .$$

**Theorem 4.4** $\mathsf{G}_1 \subseteq \mathsf{G}_2 \equiv \mathsf{M}_1 \supseteq \mathsf{M}_2$ .

**Proof**

$$\begin{array}{ll}
 & \mathsf{G}_1 \subseteq \mathsf{G}_2 \\
\equiv & \qquad \{\ \text{concept}\ \} \\
 & \mathsf{G}_1 \subseteq R\langle \mathsf{M}_2 \rangle \\
\equiv & \qquad \{\ \text{theorem 4.1}\ \} \\
 & \langle \mathsf{G}_1 \rangle R \supseteq \mathsf{M}_2 \\
\equiv & \qquad \{\ \text{concept}\ \} \\
 & \mathsf{M}_1 \supseteq \mathsf{M}_2 \ .
\end{array}$$

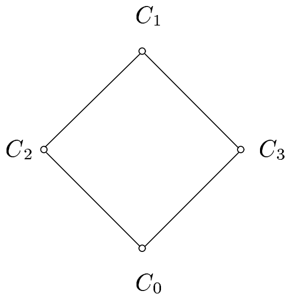


Figure 1: Concept lattice of the player

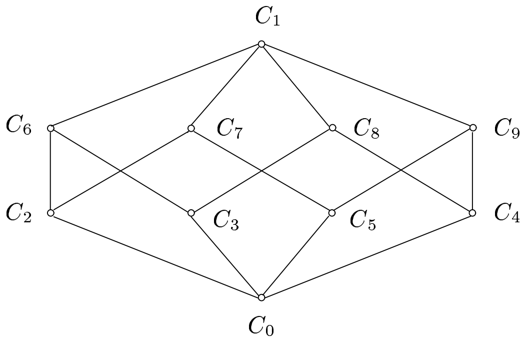


Figure 2: Concept lattice of the card-game

We illustrate concept lattice construction by two examples.

**Example 4.5 (Player, card-game)** Our first example is the "player". We specify the player as a person who plays a game against his neighbors on the left- and right-hand side. We assume that our 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 neighbors), $M_p = \{l, r\}$ ($l$ and $r$ are short for 'left' and 'right neighbors') and $R_p = \{(P_a, l), (P_a, r), (P_b, l), (P_c, r)\}$.

Two remarks are in place here. First, the above specification of the player reflects *our* 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 we have to consider each subset $\mathsf{A}$ of $G_p$ and compute the corresponding subset $\langle \mathsf{A} \rangle R_p$ of $M_p$. Let us begin with the subset $\mathsf{A} = \{P_a\}$. Since $\mathsf{A}$ is a singleton set, we get that $\langle \mathsf{A} \rangle R_p = \{l, r\}$ (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 it follows from the calculations: $R_p \langle \{l, r\} \rangle = R_p \langle \{l\} \rangle \cap R_p \langle \{r\} \rangle = \{P_a, P_b\} \cap \{P_a, P_c\} = \{P_a\}$. We may conclude that $(\{P_a\}, \{l, r\})$ is a concept.

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

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

$$
\begin{aligned}
C_0 = & \quad (\{P_a\}, \{l, r\}) , \\
C_1 = & \quad (\{P_a, P_b, P_c\}, \{\ \}) , \\
C_2 = & \quad (\{P_a, P_b\}, \{l\}) , \\
C_3 = & \quad (\{P_a, P_c\}, \{r\}) .
\end{aligned}
$$

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 get 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:

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

□

We recall the fundamental theorem on concept lattices from [1].

**Theorem 4.6** (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** $\underset{j\in J}{\sqcup}(A_j, B_j) = ((R^l.R^r.\underset{j\in J}{\cup} A_j), \underset{j\in J}{\cap} B_j)$ ,
**b** $\underset{j\in J}{\sqcap}(A_j, B_j) = (\underset{j\in J}{\cap} A_j,\ R^r.R^l.(\underset{j\in J}{\cup} B_j))$ .

(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. $L = \{\sqcup \mathsf{X} \mid \mathsf{X} \subseteq \gamma.G\}$), $\mu.M$ is meet-dense in $L$ (i.e. $L = \{\sqcap \mathsf{X} \mid \mathsf{X} \subseteq \mu.M\}$), and $gRm \equiv \gamma.g \leq \mu.m$ 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 $\underset{j\in J}{\cup} A_j$. Then, determine for this set the (Galois) closed elements, that is, $R^l.R^r.\underset{j\in J}{\cup} A_j$ which is a superset of $\underset{j\in J}{\cup} A_j$. Clearly, those objects of $\underset{j\in J}{\cup} A_j$ will occur in the closure that have some attribute in common with all other objects of the set. Those "common" attributes are given by $\underset{j\in J}{\cap} B_j$.

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 $\gamma$ and $\mu$ 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 our problem mentioned in the Introduction. Essentially, the problem is about a subsumption relation on specifications. In this section we show that the concept lattice representation of specifications offer an elegant solution to that problem. The idea is that (under certain conditions) the subsumption relation on specifications can be "translated" to 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 $a \sqcap b \in X$.

Our definition of a sublattice of a concept lattice is based on the one of the sublattice above. Since an element of such a lattice is a pair of sets, we require 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.

$$\begin{tikzcd}
\underline{\mathcal{B}}_2 \arrow[d, "h"'] & (G_2, M_2, R_2) \arrow[l, "\beta"'] \arrow[d, "\varphi"] \\
\underline{\mathcal{B}}_1 & (G_1, M_1, R_1) \arrow[l, "\beta"]
\end{tikzcd}$$

Figure 3: Concept sublattice

**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 $L_1$. We say then that $L_1$ 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 $\perp_i = (G_i^{\perp}, M_i^{\perp})$ and $\top_i = (G_i^{\top}, M_i^{\top})$. We say, $\underline{\mathcal{B}}_2$ is *compatible* with $\underline{\mathcal{B}}_1$ if $G_2^{\perp} \subseteq G_1^{\perp}$ 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 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 $\underline{\mathcal{B}}_1 = \underline{\mathcal{B}}(G_1, M_1, R_1)$ and $\underline{\mathcal{B}}_2 = \underline{\mathcal{B}}(G_2, M_2, R_2)$ be concept lattices. Then $\underline{\mathcal{B}}_2$ is a *concept sublattice* of $\underline{\mathcal{B}}_1$ if $\underline{\mathcal{B}}_1$ is homomorphic to $\underline{\mathcal{B}}_2$ by some map $h$ and there exist a context embedding $\varphi$ such that the following diagram commutes (the function to concept lattice from context, described in theorem 4.6, is denoted by the symbol $\beta$):

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

**Theorem 5.6** 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, $\underline{\mathcal{B}}_2$ be compatible with $\underline{\mathcal{B}}_1$ and let $\underline{\mathcal{B}}_1$ be homomorphic to $\underline{\mathcal{B}}_2$ by $h$. Then, an injective $\varphi$ exists.

**Proof** Due to properties of $h$, we have that some $\underline{\mathcal{B}}_s$, a sublattice of $\underline{\mathcal{B}}_1$, isomorphic to $h.\underline{\mathcal{B}}_2$, must exist. By theorem 4.6, the context of this sublattice, $(G_s, M_s, R_s)$, can be determined. From theorem 4.6 and definition 4.3 follows that any concept $(C)$ of a concept lattice has more objects and (dually) less attributes than any concept smaller than $C$. By compatibility of $\underline{\mathcal{B}}_2$ with $\underline{\mathcal{B}}_1$

we have that, $G_2 \subseteq G_s$ and $M_2 \subseteq M_s$, for some such $\underline{\mathcal{B}}_s$. This implies that an injective $\varphi$ can be defined.

Theorem 5.6 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 sum, 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, e.g. 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 $(\mathsf{A}, \mathsf{B})$ of $\underline{\mathcal{B}}_1 = \underline{\mathcal{B}}(G_1, M_1, R_1)$ by adding a new object, say $g \notin G_1$, to $\mathsf{A}$. If the set of attributes of $g$ are contained in $\mathsf{B}$ 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, $\underline{\mathcal{B}}_1 \cong \underline{\mathcal{B}}_2$ and $(\mathsf{A} \cup \{g\}, \mathsf{B})$ will be a concept of $\underline{\mathcal{B}}_2$. The second corollary is similar to the first one, except that it allows elements of a context to be removed.

**Corollary 5.7 (Extendibility)**
Let $\underline{\mathcal{B}}_1 = \underline{\mathcal{B}}(G_1,M_1,R_1)$ and $\underline{\mathcal{B}}_2 = \underline{\mathcal{B}}(G_2,M_2,R_2)$, 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 $\underline{\mathcal{B}}_1 \cong \underline{\mathcal{B}}_2$ . (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 the one of $g_1$ are equivalent. From this and theorem 4.6(2) the corollary follows immediately.

**Corollary 5.8 (Normalization)**
Let $\underline{\mathcal{B}}_1 = \underline{\mathcal{B}}(G_1,\, M_1,\, R_1)$. Then $\underline{\mathcal{B}}_2 = \underline{\mathcal{B}}(G_2,\, M_2,\, R_2)$ is called a *normalized* concept lattice of $\underline{\mathcal{B}}_1$, if $\underline{\mathcal{B}}_1 \cong \underline{\mathcal{B}}_2$, $G_1 \subseteq G_2$, $M_1 \subseteq M_2$, $R_1 \subseteq R_2$, and furthermore $R^r.G_2 = \emptyset$ and $R^l.M_2 = \emptyset$.

We call a context normalized if the corresponding concept lattice is normalized.

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 our earlier subsumption problem. In this example we will assume that an algorithm for deciding the concept sublattice relation, as to definition 5.5, exists. We describe such an algorithm in section 7. The second example contains applications of corollary 5.7 and 5.8.

**Example 5.9** Assume that the specifications and the concept lattices of the player (cf. figure 1) and the card-game (cf. figure 2) are given. We observe that definition 5.5 can be applied to these concept lattices by the following assignments:

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

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

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

In sum, we have that the concept lattice of the card-game has four sublattices, isomorphic to the concept lattice of the player:

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

We recall that the specification of the card-game contains no mentioning of its players. Nevertheless we were able to find them, due to our representation and definition 5.5. □

**Example 5.10** First, let us extend the specification of the player in the sense of corollary 5.7, e.g. by adding the pair $(P_d, l)$. One might think of such a pair as a simulation for "noise". If we construct the concept lattice of the modified context we can see 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. By corollary 5.8 we get that also in this case the shape of the concept lattice remains unchanged.

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

# 6 Repeated application of a Galois connection

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

If we have a closer look at our definition of a Galois connection (definition 3.1, 3.5 and 3.6) then we can see that the parameters of the pair of functions $(R^l, R^r)$ are $G$, $M$ and, perhaps most importantly, the relation $R$.

Our intention with "repeated application" is to apply the Galois connection $(R^l, R^r)$ to a concept lattice. By this we slightly restrict our choice on the sets $G$ and $M$. Actually, we will only consider two cases. In the first one, we choose the set of elements of a concept lattice to define $G$ and $M$ and in the second one, the set of subsets of those elements.

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

The context $(G_c, M_c, R_c)$ is theoretically interesting, but we found it not directly relevant for 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 6.2** ([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 we only consider finite sets. The special element $\{\emptyset\}$ is added for reasons of completeness of the lattice.

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

**Definition 6.3** 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 $\exists \mathsf{y} \in \mathsf{b}$ ($\mathsf{x}$ and $\mathsf{y}$ not containing the empty set) such that $\mathsf{x} R_c \mathsf{y}$ or $\mathsf{y} R_c \mathsf{x}$ holds.

We have seen that a concept lattice can represent a specification. By a repeated application of a Galois connection we may achieve that 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 6.4** 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 $B$. We also require that $B_i \not\subseteq B_j$, for all $i, j \in I$ $(i \neq j)$.

**Definition 6.5** Let $\underline{\mathcal{B}} = (B, \leq)$ be a concept lattice and $\{\underline{\mathcal{B}}_j \mid j \in J\}$ a collection of concept lattices. A *decomposition of a concept lattice* $\underline{\mathcal{B}}$ is a decomposition of $B$, $\{B_i \mid i \in I\}$, such that, for all $i \in I$: $\underline{\mathcal{B}}_i = (B_i, \leq)$ is a concept sublattice of $\underline{\mathcal{B}}$ and $\underline{\mathcal{B}}_i \cong \underline{\mathcal{B}}_j$ for some $j \in J$.

**Example 6.6** In example 5.9 we have 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 6.3.

In order to make the formulae more readable, we will introduce the prefix *with* for a named subset when used as an attribute. Eventually, by definition 6.3, we get 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 get the concepts:

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

Topologically, this lattice is isomorphic to the one of the player (cf. figure 1), but as a concept lattice it has now a completely different meaning: the card-game conceptualized as *pairs of players.* Indeed, we observe 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. $\square$

# 7 An algorithm

The most important part of a lattice subsumption algorithm is the one of lattice embedding. An implementation of $\varphi$, a context embedding, may not be difficult, in general.

**Definition 7.1** For a lattice $L = (V, \leq)$ the corresponding graph $G = (V, E)$ is such that for $a, b \in V$: $(a, b) \in E$ 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 $\triangle(x)$, the *up-set* of $x$ as $\triangle(x) = \{y \in L \mid x \sqsubseteq y\}$; and $\bigtriangledown(x)$, the *down-set* of $x$ as $\bigtriangledown(x) = \{y \in L \mid y \sqsubseteq x\}$.

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

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 7.3** ([14]) Consider the lattices $L_1$ and $L_2$ of Fig. 4. Partitioning of $V_2$ by depth yields $\{(A), (B, C), (D)\}$. Below we show the stepwise computation of the match $M = \{(A, a), (B, c), (C, b), (D, e)\}$.

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

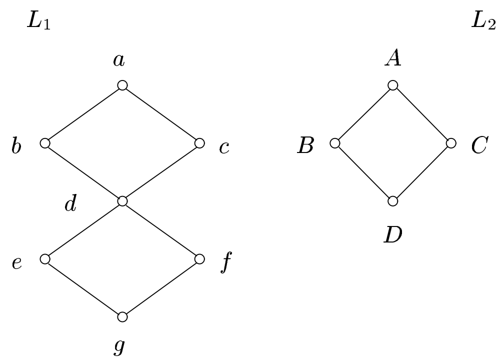

Figure 4: Sample lattices

At this point we have that all elements of $V_2$ have an image in $L_1$. The matching has to be verified for being 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. $\square$

The complexity of lattice embedding is exponential ([14]). The algorithm can be optimized by only 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 concept lattices and not some other formalism, like Prolog? The question is proper, so let us see how our running example might look like in that language. The following specification is not complete, we 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 ≤`i`≤ 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,
```

**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**
  $M_k := \{b \mid (a, b) \in M\}$;
  $\bigtriangledown(M_k) := \{a \mid \exists_{c \in M_k} : (a, c) \in E_1^* \text{ and } a \notin M_k\}$;
  **for** *every matching* $m$ *between* $C_k$ *and*
   *a subset* $X$ *of* $\bigtriangledown(M_k)$ *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

```
  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 we have not considered the implementation issues of concept lattices. 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 have complete and correct. Concept lattices allow more systematic specifications. This is, amongst others, why we think our 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. We apply our theory to a pair of specifications and make a comparison with another approach that is based on Prolog.

Our current research in the area of concept lattices includes: (i) applying concept sublattices in transformational program specification, (ii) and 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 modeled by using concept lattices, and how such a model can help us 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. (1982) Restructuring lattice theory: an approach based on hierarchies of concepts. In I. Rival (editor), *Ordered sets*, 445–470. D. Reidel, Dordrecht-Boston.

[2] Sarbo, J. and Farkas, J. (1994) Concept sublattices. In *Proc. of the European Conference on Machine Learning (ECML'94)*, Catania, Italy.

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

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

[5] Godin, R., Missaoui R. and Alaoui, H. (1995) Incremental concept formation algorithms based on Galois (concept) lattices. *Computational Intelligence*, **11**(2), 246–267.

[6] Godin, R. and Missaoui, R. (1994) An incremental concept formation approach for learning from databases. *Theoretical Computer Science*, **133**, 387–419.

[7] Aart, C., et. al. (1992) *A relational theory of datatypes*. In preparation: available by ftp from `ftp.win.tue.nl`.

[8] Birkhoff, G. (1967) *Lattice theory*. American Mathematical Society.

[9] Ganter, B. and Wille, R. (1989) Conceptual scaling. In F. Roberts (editor), *Applications of combinatorics and graph theory to the biological and social sciences*, 139–167, Springer-Verlag.

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

[11] Grice, H.P. (1975) Logic and conversation. In P. Cole and J. Morgan (eds), *Syntax and semiotics: Speech acts*, **3**, 41–58, New York, Academic Press.

[12] Sarbo, J.J. and Farkas, J.I. (1995) Knowledge representation and acquisition by concept lattices. In Shaul Markovitch (editor), *Proc. of the 11th Izraeli Symposium on Artificial Intelligence (ISAI'95)*, Hebrew University of Jerusalem, Izrael.

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

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