Representations of Information Systems and Dependences Spaces, and Some Basic Algorithms

LICENTIATE'S THESIS

Jouni Järvinen

University of Turku
Department of Mathematics
FIN-20014 TURKU

April 1997

Contents

Chapter 1

Introduction

According to Z. Pawlak knowledge about a universe of objects may be defined as classifications based on certain properties of the objects. In this work we concentrate merely on such classifications which form a partition of the given object set, that is, each object belongs to exactly one category. Because it is well-known that the relationship between partitions and equivalence relations is bijective, knowledge about objects may as well be given in terms of equivalence relations. Any equivalence e can now be interpreted as an *indiscernibility relation* which satisfies $(x, y) \in e$ if we cannot discern objects x and y by the knowledge e. For example, if we classify all human beings into two disjoint sets consisting of women and men, then this classification determines an equivalence relation e in the set of people such that $(x, y) \in e$ whenever x and y are of the same sex. Note, that equivalence relations are reflexive and symmetric, which are intuitively quite natural requirements for indiscernibility. Transitivity is not a so obvious property, and in e.g. [15, 22] *similarity relations* which are only reflexive and symmetric are considered.

By a *knowledge base* we understand a pair $\mathcal{K} = (U, E)$, where U is a nonempty set of objects and E is a set of equivalences on U (see [21], for example). We can derive new knowledge about objects of a knowledge base by applying the set-theoretical operation intersection to subsets of E. Each subset $D (\subseteq E)$ defines an *indiscernibility relation* $Ind(D) = \bigcap D$ on U such that $(x, y) \in Ind(D)$ if and only if x and y are indiscernible with respect to all $e \in D$. In the study of the structure of the set of all indiscernibility relations defined by subsets of E the notions of indispensable elements, independent subsets, cores, reducts, and dependency relations have important roles.

Pawlak introduced the notion of information systems (sometimes called knowledge representation systems) in [19]. Information concerning properties of objects is the basic knowledge included in information systems and it is given in terms of attributes and values of attributes. For example, we may express statements concerning the color of objects if the information system includes an attribute “color” and a set of values of this attribute consisting of “yellow”, “green” etc.

In general, an *information system* is determined by specifying a set of *objects* U, a set A of *attributes* meaningful for all objects, and for every attribute $a \in A$, a fixed set V_a of *values* of that attribute. Here we assume that the basic information of objects is single-valued and completely defined. Therefore, every attribute a can be considered as a total mapping $a : U \to V_a$ which assigns to each object $x \in U$ the unique value

$a(x)$ of the attribute a.

It is well-known that the kernel of a total mapping is an equivalence. Hence, in an information system $\mathcal{S} = (U, A, \{V_a\}_{a \in A})$ for any $a \in A$ the kernel of a, defined by $(x, y) \in \ker a$ if and only if $a(x) = a(y)$, is an equivalence on U. We may now view each equivalence relation $\ker a$ as an indiscernibility relation because $(x, y) \in \ker a$ whenever the objects x and y are indiscernible with respect to the attribute a. Therefore, an information system $\mathcal{S}$ defines a knowledge base. Namely, if we set $E_{\mathcal{S}} = \{\ker a \mid a \in A\}$, then the pair $\mathcal{K}_{\mathcal{S}} = (U, E_{\mathcal{S}})$ is obviously a knowledge base. Each set $B(\subseteq A)$ of attributes defines now an indiscernibility relation $Ind(B) = \bigcap_{a \in B} \ker a$.

In the theory of information systems there are two major problems. The first is usually referred to as the *reduction problem* and it is stated as follows. Suppose B is a subset of attributes of an information system. We have to find the set of all minimal subsets C of B which satisfy $Ind(B) = Ind(C)$. The other important problem is associated with dependency relations. A subset of attributes B is *dependent on* a subset C of attributes in $\mathcal{S}$, denoted by $C \to B\ (\mathcal{S})$, if $Ind(C) \subseteq Ind(B)$. This means simply that the values for the attributes in B can be determined from the values for the attributes in C. The problem is to find for a dependency $C \to B\ (\mathcal{S})$ the set of all minimal subsets D of C which satisfy $D \to B\ (\mathcal{S})$.

We present a solution to the first problem by applying discernibility matrices and discernibility functions (see e.g. [26]). In addition to this we introduce dependency functions, and by means of these functions we solve the latter problem in a way which differs essentially from the solution presented in [26]. We have found out that these two problems can be reduced to the general problem of identifying the set of all minimal true vectors of a isotone Boolean function (see [9], for example).

An another important topic of this work is the theory of dependence spaces. A *depend-ence space* (as defined by Novotný and Pawlak) is a pair $\mathcal{D} = (A, K)$ where A is a finite set and K is a congruence on the semilattice $(\wp(A), \cup)$, where $\wp(A)$ denotes the set of all subsets of A. If we define for an information system $\mathcal{S} = (U, A, \{V_a\}_{a \in A})$, in which the set A is finite, a binary relation $K_{\mathcal{S}}$ on $\wp(A)$ by setting

$$(B, C) \in K_{\mathcal{S}} \text{ if and only if } Ind(B) = Ind(C)$$

for all $B, C \subseteq A$, then it can be easily verified that the pair $\mathcal{D}_{\mathcal{S}} = (A, K_{\mathcal{S}})$ is a dependence space.

It is known that many problems concerning information systems can be formulated in the the more abstract setting of dependence spaces. It has also been proved that dependence spaces provide a suitable basis for the study of several problems concerning contexts (in the sense of Wille) (see e.g. [4, 11]).

Because for a dependence space $\mathcal{D} = (A, K)$ the set $\wp(A)$ is finite and it contains the least element $\emptyset$, it is easy to observe that the quotient semilattice $(\wp(A)/K, \leq)$ is always a finite lattice and hence complete. Moreover, it is isomorphic to the complete lattice $(\mathcal{L}_{\mathcal{D}}, \subseteq)$ where $\mathcal{L}_{\mathcal{D}}$ is a closure system which corresponds to the closure operator $\mathcal{C}_{\mathcal{D}} : \wp(A) \to \wp(A),\ B \mapsto \bigcup B/K$. We have found out that any dependence space $\mathcal{D}$ can be characterized by a subset $\mathcal{T} \subseteq \wp(A)$ which satisfies $\mathcal{M}(\mathcal{L}_{\mathcal{D}}) \subseteq \mathcal{T} \subseteq \mathcal{L}_{\mathcal{D}}$, where $\mathcal{M}(\mathcal{L}_{\mathcal{D}})$ denotes the set of meet-irreducible elements of the lattice $\mathcal{L}_{\mathcal{D}}$ which differ from A. Such sets $\mathcal{T}$ are called *dense* (see e.g. [14]).

The notions of indispensable elements, independent subsets, cores, and dependency relations play important roles also in the theory of dependence spaces. Especially, we are interested in finding a solution to the reduction problem in dependence spaces, i.e., for an arbitrary subset $B(\subseteq A)$ of a dependence space we want to enumerate all minimal subsets C of B which satisfy $(B, C) \in K$. We have found a convenient new way to characterize the reducts of an arbitrary subset by means of dense sets. The dependency relation is defined in dependence space $\mathcal{D}$ in terms of closure operator $\mathcal{C}_{\mathcal{D}}$. A subset B is *dependent on* a subset C in $\mathcal{D}$, denoted by $C \to B$ $(\mathcal{D})$, whenever $\mathcal{C}_{\mathcal{D}}(B) \subseteq \mathcal{C}_{\mathcal{D}}(C)$ holds. We shall present for a dependency $C \to B$ $(\mathcal{D})$ a method that identifies the set of all minimal subsets D of C which satisfy $D \to B$ $(\mathcal{D})$. In addition, we shall introduce the notions of difference and dependency functions, which are somewhat similar to the discernibility and dependency functions in the case of information systems. Finally, we shall present an algorithm which converts in a polynomial time the representation of an information system $\mathcal{S}$ to the representation of the corresponding dependence space $\mathcal{D}_{\mathcal{S}}$.

This work is structured as follows. In the following chapter we recall some notions and notation of lattice theory and universal algebra. In Chapter 3 we present a generalized version of knowledge bases, and study especially the cores and reducts of subsets in knowledge bases in which the sets U and E are infinite. Chapter 4 is devoted to the study of information systems, and especially the structure of the complete lattice of all indiscernibility relations is considered. In Chapter 5 we investigate discernibility matrices, discernibility functions, and dependency functions. Moreover, we present some algorithms. In Chapter 6 we first study closure operators and dense sets in finite semilattices which have a zero element, and then apply our results to dependence spaces. The relationship between information systems and dependence spaces is also considered. Finally, in Chapter 7 we introduce difference and dependency functions, and present some of their important properties. This chapter contains also a selection algorithms concerning dependence spaces.

This study was carried out at the Department of Mathematics, University of Turku, and it has been financially supported by the Turku Centre for Computer Science/Graduate School. I am grateful to Professor Magnus Steinby for his expert guidance and valuable suggestions during the course of this work.

Chapter 2

Preliminaries

2.1 Sets

All general lattice theoretical and algebraic notions used in this work can be found in [2, 3, 4, 7], for example. We assume that the reader is familiar with the following notations: *set-builder* ($\{- \mid -\}$), *membership* ($\in$), *subset* ($\subseteq$), *proper subset* ($\subset$), *union* ($\cup$), *intersection* ($\cap$), *difference* ($-$), *ordered n-tuples* ($(x_1, \ldots, x_n)$), and *products of sets* ($A_1 \times \cdots \times A_n$). The notations A_i, $i \in I$, and $\{A_i\}_{i \in I}$ refer to a *family of sets indexed by a set I*. Given a family F of sets, the union of F, $\bigcup F$, is defined by $a \in \bigcup F$ if and only if $a \in A$ for some $A \in F$. The *intersection* of F, $\bigcap F$, is defined dually. For a set A, let $\wp(A)$ denote the *power set* of A, that is, the set of all subsets of A.

Let us write $\mathbb{N} = \{1, 2, \ldots\}$ and $\mathbb{N}_0 = \{0\} \cup \mathbb{N}$. The *cardinal* of the set A is denoted by $|A|$. A set A is said to be *finite* if $|A| = n$ for some $n \in \mathbb{N}_0$; otherwise A is *infinite*. In particular the empty set, $\emptyset$, is finite and its cardinal is 0.

2.2 Relations and functions

An *n-ary relation* r on a set A is a subset of A^n. If $n = 2$, then r is called a *binary relation*. We denote by $\mathrm{Rel}(A)$ the set of all binary relations in the set A. For all $r \in \mathrm{Rel}(A)$ the relation $r^{-1} = \{(a, b) \mid (b, a) \in r\}(\in \mathrm{Rel}(A))$ is called the *inverse* of r. A relation $r \in \mathrm{Rel}(A)$ is:

- *reflexive*, if for all $a \in A$, $(a, a) \in r$;
- *symmetric*, if for all $a, b \in A$, $(a, b) \in r$ implies $(b, a) \in r$;
- *antisymmetric*, if for all $a, b \in A$, $(a, b) \in r$ and $(b, a) \in r$ imply $a = b$;
- *transitive*, if for all $a, b, c \in A$, $(a, b) \in r$ and $(b, c) \in r$ imply $(a, c) \in r$.

A binary relation is an *equivalence relation* if it is reflexive, symmetric, and transitive. We denote by $\mathrm{Eq}(A)$ the set of all equivalence relations on A. If $e \in \mathrm{Eq}(A)$ and $a \in A$, then the *equivalence class of a modulo e* is the set $a/e = \{b \in A \mid (a, b) \in e\}$. The *quotient set of A modulo e* is the set $A/e = \{a/e \mid a \in A\}$.

A *partition* π of a set A is a family of nonempty pairwise disjoint subsets of A such that $A = \bigcup \pi$. The sets in π are called the *blocks* of π. The set of all partitions of

A is denoted by $\Pi(A)$. If $e \in \mathrm{Eq}(A)$, then A/e is a partition of A. For any partition $\pi \in \Pi(A)$, there exists a unique equivalence e_π such that $A/e_\pi = \pi$; e_π is defined by $(a, b) \in e_\pi$ if and only if $\{a, b\} \subseteq B$ for some $B \in \pi$.

A *function* (or a *mapping*) f from a set A to a set B, denoted $f : A \to B$, is a subset of $A \times B$ such that for each $a \in A$, there exists exactly one $b \in B$ with $(a, b) \in f$; in which case we write $f(a) = b$ or $f : a \mapsto b$. The set of all functions from A to B is denoted by B^A. Suppose $f \in B^A$. Then f is *injective* (or *one-to-one*) if $f(a_1) = f(a_2)$ implies $a_1 = a_2$. The function f is *surjective* (or *onto*) if for every $b \in B$, there exists an element $a \in A$ with $f(a) = b$. Further, f is *bijective* if it is both injective and surjective.

For $f : A \to B$ and $g : B \to C$, let $g \circ f : A \to C$ be the function defined by $(g \circ f)(a) = g(f(a))$. The function $g \circ f$ is called the *product* of functions g and f. A function $1_A : A \to A, a \mapsto a$, is called the *identity function* of A. A function $g : B \to A$ is the *inverse function* of $f : A \to B$ if $g \circ f = 1_A$ and $f \circ g = 1_B$. It is known that $f : A \mapsto B$ has an inverse function if and only if f is a bijection. The inverse of a bijection f is denoted by f^{-1}.

If A is a set and $e \in \mathrm{Eq}(A)$, then the function $v_e : A \to A/e, a \mapsto a/e$ is called the *canonical map* of e. Obviously, the function v_e is surjective. The *kernel* of the function $f : A \to B$ is a relation $\ker f (\in \mathrm{Rel}(A))$ defined by $(a, b) \in \ker f$ if and only if $f(a) = f(b)$, for all $a, b, \in A$. It is easy to verify that $\ker f \in \mathrm{Eq}(A)$.

2.3 Ordered sets and lattices

Suppose P is a set. An *order* (or a *partial order*) on P is a binary relation $\leq$ such that, for all $a, b, c \in P$, (i) $a \leq a$, (ii) $a \leq b$ and $b \leq a$ imply $a = b$, (iii) $a \leq b$ and $b \leq c$ imply $a \leq c$, that is, the relation $\leq$ is reflexive, antisymmetric, and transitive. A set P equipped with an order relation $\leq$ is said to be an *ordered set* (or a *partially ordered set*). Some authors use the shorthand *poset*.

Let $(P, \leq)$ be an ordered set and let $a, b \in P$. We say a is *covered by* b (or b *covers* a), and write $a \!-\!\!< b$, if $a < b$ and $a \leq c < b$ implies $a = c$. The latter condition requires that there is no element c of P which would satisfy $a < c < b$. Note, that if P is finite, then $a < b$ if and only if there exists a finite sequence of covering relations $a = a_0 \!-\!\!< a_1 \!-\!\!< \cdots \!-\!\!< a_n = b$. Thus, in the finite case, the order relation is determined by the covering relation.

If P and Q are ordered sets, then a mapping $\varphi : P \to Q$ is an *order-isomorphism*, if $a \leq b$ in P if and only if $\varphi(a) \leq \varphi(b)$ in Q and φ is bijective. When there exists an order-isomorphism from P to Q, we say that P and Q are *order-isomorphic* and write $P \cong Q$.

Let $(P, \leq)$ be an ordered set. Then P is a *chain* if, for all $a, b \in P$, either $a \leq b$ or $b \leq a$, that is, any two elements of P are comparable. The ordered set P is an *antichain* if $a \leq b$ in P only if $a = b$. Suppose $S \subseteq P$. Then $a \in S$ is a *maximal* element of S, if $a \leq x \in S$ implies $a = x$. The set of all maximal elements in S is denoted by $\max S$. Further, $a \in S$ is the *greatest* element of S, if $x \leq a$ for all $x \in S$. The set of *minimal* elements, $\min S$, and the *least* element of S are defined dually, that is, by reversing the order.

The greatest element of P, if it exists, is called the *top element* of P and written

$\top$. Similarly, the least element of P, if such exists, is called the *bottom element* and it is denoted by $\bot$. If $S \subseteq P$, then an element $x \in P$ is an *upper bound* of S if $a \leq x$ for all $a \in S$. A *lower bound* is defined dually. The set of all upper bounds of S is denoted by S^u and the set of all lower bounds by S^l.

If S^u has a least element, this is called the *least upper bound* of S. Dually, if S^l has a greatest element, this is called the *greatest lower bound* of S. The least upper bound of S is also called the *supremum* of S and is denoted by $\sup S$. Similarly, the greatest lower bound of S is also called the *infimum* of S and is denoted by $\inf S$.

We write $a \vee b$ (read as "*a join b*") in place of $\sup\{a, b\}$ and $a \wedge b$ (read as "*a meet b*") in place of $\inf\{a, b\}$. Similarly, we write $\bigvee S$ and $\bigwedge S$ instead of $\sup S$ and $\inf S$, respectively. Obviously, $\emptyset^u = P$ and $\bigvee \emptyset$ exists if and only if P has a bottom element $\bot$, and in that case $\bigvee \emptyset = \bot$. Dually, $\bigwedge \emptyset = \top$ whenever P has a top element.

A nonempty ordered set $(P, \leq)$ is a *lattice* if $a \vee b$ and $a \wedge b$ exist for all $a, b \in P$. If $\bigvee S$ and $\bigwedge S$ exist for all $S \subseteq P$, then $(P, \leq)$ is called a *complete lattice*. To show that an ordered set is a complete lattice requires only half as much work as the definitions would have us to believe.

Lemma 2.1 *Let P be an ordered set such that $\bigwedge S$ exists in P for every nonempty subset S of P. Then $\bigvee S$ exists in P for every nonempty subset S of P which has an upper bound in P; indeed, $\bigvee S = \bigwedge S^u$.* □

Example 2.2 Assume A is a set. It is clear that $(\wp(A), \subseteq)$ is a ordered set in which $\bot = \emptyset$ and $\top = A$. Moreover, $(\wp(A), \subseteq)$ is a complete lattice with

$$\bigvee\{B_i \mid i \in I\} = \bigcup_{i \in I} B_i, \quad \text{and}$$
$$\bigwedge\{B_i \mid i \in I\} = \bigcap_{i \in I} B_i.$$

Because $\mathrm{Rel}(A) = \wp(A \times A)$, it follows from the above that $(\mathrm{Rel}(A), \subseteq)$ is a complete lattice with join given by set union and meet given by set intersection.

Lemma 2.3 *Let $(P, \leq)$ an ordered set, let $S, T \subseteq P$, and assume $\bigvee S$, $\bigvee T$, $\bigwedge S$, and $\bigwedge T$ exist in P. If $S \subseteq T$, then $\bigvee S \leq \bigvee T$ and $\bigwedge T \leq \bigwedge S$.* □

We say P satisfies the *ascending chain condition* (ACC), if given any sequence $a_1 \leq a_2 \leq \cdots \leq a_n \cdots$ of elements of P, there exists a $k \in \mathbb{N}$ such that $a_k = a_{k+1} = \cdots$. The dual of the ascending chain condition is the *descending chain condition* (DCC). It is obvious that every finite ordered set satisfies both the ACC and the DCC. The following lemma is very useful.

Lemma 2.4 *An ordered set P satisfies the* ACC *if and only if every nonempty subset S of P has a maximal element.* □

2.4 Closure operators and closure systems

A system $\mathcal{L}$ of subsets of A is said to be a *closure system* if $\mathcal{L}$ is closed under intersections, i.e., for all subsystems $\mathcal{H} \subseteq \mathcal{L}$, we have $\bigcap \mathcal{H} \in \mathcal{L}$. A *closure operator* on a set A is an extensive, idempotent, and isotone function $\mathcal{C} : \wp(A) \to \wp(A)$, that is,

(a) $B \subseteq \mathcal{C}(B)$,
(b) $\mathcal{C}(\mathcal{C}(B)) = \mathcal{C}(B)$, and
(c) $B \subseteq C$ implies $\mathcal{C}(B) \subseteq \mathcal{C}(C)$

for all $B, C \subseteq A$. A subset B of A is *closed* (with respect to $\mathcal{C}$) if $\mathcal{C}(B) = B$. A closure system $\mathcal{L}$ on A defines a closure operator $\mathcal{C}_{\mathcal{L}}$ on A by the rule

$$\mathcal{C}_{\mathcal{L}}(B) = \bigcap \{L \in \mathcal{L} \mid B \subseteq L\}.$$

Conversely, if $\mathcal{C}$ is a closure operator on A, then the family

$$\mathcal{L}_{\mathcal{C}} = \{B \subseteq A \mid \mathcal{C}(B) = B\}$$

of closed subsets of A is a closure system. The relationship between closure systems and closure operators is bijective. The closure operator induced by the closure system $\mathcal{L}_{\mathcal{C}}$ is $\mathcal{C}$ itself, and similarly the closure system induced by the closure operator $\mathcal{C}_{\mathcal{L}}$ is $\mathcal{L}$. In symbols,

$$\mathcal{C}_{(\mathcal{L}_{\mathcal{C}})} = \mathcal{C} \quad \text{and} \quad \mathcal{L}_{(\mathcal{C}_{\mathcal{L}})} = \mathcal{L}.$$

Suppose $\mathcal{L}$ is a closure system on A. Clearly, the ordered set $(\mathcal{L}, \subseteq)$ has the top element $\top = \bigcap \emptyset = A$ and the bottom element $\bot = \bigcap \mathcal{C}$. Further, the ordered set $(\mathcal{L}, \subseteq)$ is a complete lattice. If $\{B_i \mid i \in I\}$ is a nonempty subset of $\mathcal{L}$, then

$$\begin{aligned}
\bigwedge \{B_i \mid i \in I\} &= \bigcap_{i \in I} B_i; \\
\bigvee \{B_i \mid i \in I\} &= \bigwedge \{B_i \mid i \in I\}^u \\
&= \bigcap \{L \in \mathcal{L} \mid B_i \subseteq L \text{ for all } i \in I\} \\
&= \bigcap \{L \in \mathcal{L} \mid \bigcup_{i \in I} B_i \subseteq L\} \\
&= \mathcal{C}_{\mathcal{L}}(\bigcup_{i \in I} B_i).
\end{aligned}$$

Example 2.5 Suppose A is a set. Let us consider the set $\mathrm{Eq}(A)$ of all equivalences on A. If $\{e_i \mid i \in I\}$ is a nonempty subset of $\mathrm{Eq}(A)$, then clearly $\bigcap_{i \in I} e_i \in \mathrm{Eq}(A)$. Moreover, $\bigcap \emptyset = A \times A \in \mathrm{Eq}(A)$. Hence, the set $\mathrm{Eq}(A)$ is a closure system on $A \times A$. The corresponding closure operator is a function $\mathcal{C}_{\mathrm{Eq}} : \mathrm{Rel}(A) \to \mathrm{Rel}(A)$. It returns for all $r \in \mathrm{Rel}(A)$ the smallest equivalence relation on A containing r.

The ordered set $\mathrm{Eq}(A)$ of all equivalence relations on A is a complete lattice. Suppose $\{e_i \mid i \in I\}$ is a nonempty subset of $\mathrm{Eq}(A)$. Then

$$\begin{aligned}
\bigwedge \{e_i \mid i \in I\} &= \bigcap_{i \in I} e_i\,; \\
\bigvee \{e_i \mid i \in I\} &= \mathcal{C}_{\mathrm{Eq}}(\bigcup_{i \in I} e_i).
\end{aligned}$$

Lemma 2.6 *If $\mathcal{C}$ is a closure operator on A, then the following facts hold for all $B, C \subseteq A$ and $a \in A$.*

(a) *$\mathcal{C}(B \cup C) = \mathcal{C}(\mathcal{C}(B) \cup \mathcal{C}(C))$ for all $B, C \subseteq A$.*
(b) *$a \in \mathcal{C}(B)$ if and only if $\mathcal{C}(B) = \mathcal{C}(B \cup \{a\})$.*
(c) *For all $L \in \mathcal{L}_{\mathcal{C}}$, $B \subseteq L$ if and only if $\mathcal{C}(B) \subseteq L$.*

Proof. (a) $\mathcal{C}(B \cup C) \subseteq \mathcal{C}(\mathcal{C}(B) \cup \mathcal{C}(C)) \subseteq \mathcal{C}(\mathcal{C}(B \cup C)) = \mathcal{C}(B \cup C)$. (b) If $a \in \mathcal{C}(B)$, then $\mathcal{C}(B) \subseteq \mathcal{C}(B \cup \{a\}) \subseteq \mathcal{C}(\mathcal{C}(B)) = \mathcal{C}(B)$. If $a \notin \mathcal{C}(B)$, then $\mathcal{C}(B) \subset \mathcal{C}(B) \cup \{a\} \subseteq \mathcal{C}(B \cup \{a\})$. (c) Suppose $L \in \mathcal{L}_{\mathcal{C}}$. If $B \subseteq L$, then $\mathcal{C}(B) \subseteq \mathcal{C}(L) = L$. On the other hand, $\mathcal{C}(B) \subseteq L$ trivially implies $B \subseteq L$. ☐

2.5 Algebras, homomorphisms, and congruences

For a nonempty set A and a nonnegative integer n, we define $A^0 = \{\emptyset\}$ and for $n > 0$, A^n is the set of n-tuples of elements from A. An *n-ary operation* (or *function*) on A is any function f from A^n to A; n is the *arity* (or rank) of f. A *finitary operation* is an n-ary operation for some n. The image of $(a_1, \ldots, a_n)$ under an n-ary operation f is denoted by $f(a_1, \ldots, a_n)$. A function f on A is called a *constant* if its arity is zero. It is completely determined by the image $f(\emptyset)$ in A. Hence, it is convenient to identify it with this element of A. An operation f on A is *unary, binary* or *ternary* if its arity is 1, 2, or 3, respectively.

A *language* (or *type*) of algebras is a set Σ of *function symbols* such that a nonnegative integer n is assigned to each member f of Σ. This integer is called the *arity* (or *rank*) of f, and f is said to be an *n-ary function symbol*. The subset of n-ary function symbols in Σ is denoted by Σ_n.

Let A be a set and Σ a set of function symbols. A Σ-*algebra* is an ordered pair $\mathcal{A} = (A, F)$ where F is a family of finitary operations of A indexed by the language Σ such that corresponding to each n-ary function symbol $f \in \Sigma$ there is an n-ary operation $f^{\mathcal{A}}$ on A. The set A is called the *universe* of $\mathcal{A}$ and the $f^{\mathcal{A}}$'s are called the *fundamental operations of* $\mathcal{A}$. Usually we write (A, Σ) instead of (A, F). Furthermore, sometimes we mean by the type of a Σ-algebra a list of the arities of the function symbols in Σ. Also we often drop the upper index from $f^{\mathcal{A}}$.

Assume $\mathcal{A} = (A, \Sigma)$ and $\mathcal{B} = (B, \Sigma)$ are Σ-algebras. A function $\varphi : A \to B$ is a *homomorphism* from algebra $\mathcal{A}$ to algebra $\mathcal{B}$, denoted by $\varphi : \mathcal{A} \to \mathcal{B}$, if for every n-ary $f \in \Sigma$ and $a_1, \ldots, a_n$, we have

$$\varphi(f^{\mathcal{A}}(a_1, \ldots, a_n)) = f^{\mathcal{B}}(\varphi(a_1), \ldots, \varphi(a_n)).$$

A homomorphism $\varphi : \mathcal{A} \to \mathcal{B}$ is

- an *embedding* (or *monomorphism*), if it is injective;
- an *epimorphism*, if it is surjective;
- an *isomorphism*, if it is bijective.

We say that $\mathcal{A}$ is *isomorphic* to $\mathcal{B}$, denoted by $\mathcal{A} \cong \mathcal{B}$, if there is an isomorphism from $\mathcal{A}$ to $\mathcal{B}$.

Lemma 2.7 *The product of homomorphisms is again a homomorphism, and similar statements apply for embeddings, epimorphisms, and isomorphisms. Furthermore, the inverse of an isomorphism is an isomorphism.* ☐

Let $\mathcal{A} = (A, \Sigma)$ and $\mathcal{B} = (B, \Sigma)$ be two algebras. Then $\mathcal{B}$ is a *subalgebra* of $\mathcal{A}$ if $B \subseteq A$ and every fundamental operation of $\mathcal{B}$ is the restriction of the corresponding operation of $\mathcal{A}$, i.e., for all function symbols $f \in \Sigma$, $f^{\mathcal{B}}$ is $f^{\mathcal{A}}$ restricted to B. A *subuniverse* of A is a subset B of A, which is closed under the operations of $\mathcal{A}$, that is, for all $n \in \mathbb{N}_0$, $f \in \Sigma_n$ and $a_1, \ldots, a_n \in B$, $f^{\mathcal{A}}(a_1, \ldots, a_n) \in B$. The relationship between nonempty subuniverses of an algebra and its subalgebras is bijective:

- If $\mathcal{B}$ is a subalgebra of $\mathcal{A}$, then B is a subuniverse of $\mathcal{A}$.
- If B is a subuniverse of $\mathcal{A}$ and $B \neq \emptyset$, then we get a subalgebra $\mathcal{B} = (B, \Sigma)$ of $\mathcal{A}$ by restricting the operations of $\mathcal{A}$ in B.

Lemma 2.8 *If $\varphi : \mathcal{A} \to \mathcal{B}$ is a homomorphism, then the image of a subuniverse of $\mathcal{A}$ under φ is a subuniverse of $\mathcal{B}$.* □

Let $\mathcal{A} = (A, \Sigma)$ be a Σ-algebra and let $K \in \mathrm{Eq}(A)$. Then K is a *congruence* on $\mathcal{A}$ if K satisfies for each n-ary function symbol $f \in \Sigma$ and any elements $a_1, \ldots, a_n$,

$$\text{if } (a_i, b_i) \in K \text{ holds for } 1 \leq i \leq n, \text{ then } (f^{\mathcal{A}}(a_1, \ldots, a_n), f^{\mathcal{A}}(b_1, \ldots, b_n)) \in K.$$

The set of all congruences on an algebra $\mathcal{A}$ is denoted by $\mathrm{Con}(\mathcal{A})$. If K is a congruence on an algebra $\mathcal{A}$, then the *quotient algebra of $\mathcal{A}$ modulo K*, denoted by $\mathcal{A}/K$, is the algebra whose universe is A/K and whose fundamental operations satisfy

$$f^{\mathcal{A}/K}(a_1/K, \ldots, a_n/K) = f^{\mathcal{A}}(a_1, \ldots, a_n)/K$$

where $a_1, \ldots, a_n \in A$ and f is an n-ary function symbol in Σ. We note that the quotient algebras of $\mathcal{A}$ are of the same type as $\mathcal{A}$.

Lemma 2.9 *If $\varphi : \mathcal{A} \to \mathcal{B}$ is a homomorphism, then following facts hold.*

(a) *The kernel of φ,* $\ker \varphi$, *is a congruence of $\mathcal{A}$.*

(b) *If K is a congruence of $\mathcal{A}$, then the canonical map v_k from $\mathcal{A}$ to the quotient algebra $\mathcal{A}/K$ is an epimorphism.* □

In the literature the following Homomorphism Theorem is also referred to as "The First Isomorphism Theorem".

Theorem 2.10 (Homomorphism Theorem) *Suppose $\varphi : \mathcal{A} \to \mathcal{B}$ is a homomorphism onto B. Then there is an isomorphism ψ from $\mathcal{A}/\ker\varphi$ to $\mathcal{B}$ such that $\varphi = \psi \circ v$, where v is the canonical map from $\mathcal{A}$ to $\mathcal{A}/\ker\varphi$* (see Figure 1). □

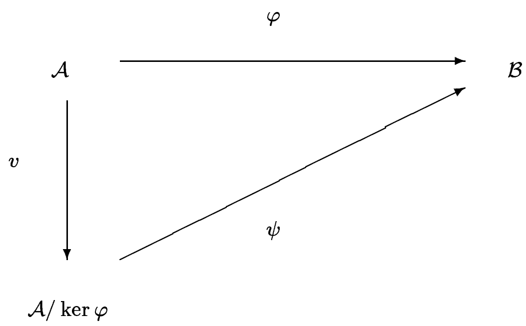

Figure 1

2.6 Lattices as algebras

In Section 2.3 we saw that for a lattice L we may define the binary operations join and meet on L by

$$a \vee b = \sup\{a, b\} \quad \text{and} \quad a \wedge b = \inf\{a, b\}$$

for all $a, b \in L$. In this section we study the algebraic properties of the operations $\vee$ and $\wedge$.

Lemma 2.11 (Connecting Lemma) *Let $(L, \leq)$ be a lattice and let $a, b \in L$. Then the following are equivalent.*
(a) $a \leq b$.
(b) $a \vee b = b$.
(c) $a \wedge b = a$. □

Theorem 2.12 *Let $(L, \leq)$ be a lattice. Then $\vee$ and $\wedge$ satisfy for all $a, b, c \in L$,*

(L1) $(a \vee b) \vee c = a \vee (b \vee c)$
(L1)$^\partial$ $(a \wedge b) \wedge c = a \wedge (b \wedge c)$ (associative laws)
(L2) $a \vee b = b \vee a$
(L2)$^\partial$ $a \wedge b = b \wedge a$ (commutative laws)
(L3) $a \vee a = a$
(L3)$^\partial$ $a \wedge a = a$ (idempotency laws)
(L4) $a \vee (a \wedge b) = a$
(L4)$^\partial$ $a \wedge (a \vee b) = b$ (absorption laws)

□

We say that an algebra $(L, \vee, \wedge)$ is a *lattice*, if L is nonempty set and $\vee$ and $\wedge$ are binary operations on L which satisfy (L1)–(L4) and (L1)$^\partial$–(L4)$^\partial$.

If an ordered set $(L, \leq)$ is a lattice, then by Theorem 2.12 the algebra $(L, \vee, \wedge)$ is a lattice. Similarly, if an algebra $(L, \vee, \wedge)$ is a lattice and we define $a \leq b$ if and only if $a \vee b = b$ for all $a, b \in L$, then the ordered set $(L, \leq)$ is a lattice in which the the original operations agree with the induced operations, that is, $a \vee b = \sup\{a, b\}$ and $a \wedge b = \inf\{a, b\}$.

Let $(L, \vee, \wedge)$ be a lattice. We say L has a *unit* (or *identity*) element if there exists $1 \in L$ such that $a \wedge 1 = a$ for all $a \in L$. Dually, L is said to have a *zero* if there exists $0 \in L$ such that $a = a \vee 0$ for all $a \in L$. The lattice $(L, \vee, \wedge)$ has a unit if and only if $(L, \leq)$ has a top element $\top$ and in that case $1 = \top$. A dual statement holds for 0 and $\bot$. A lattice $(L, \vee, \wedge)$ possessing 0 and 1 is called *bounded*. A finite lattice is automatically bounded, with $1 = \bigvee L$ and $0 = \bigwedge L$.

Let $\mathcal{L} = (L, \vee, \wedge)$ be a lattice. If $\emptyset \neq S \subseteq L$ is a subuniverse of $\mathcal{L}$, then $(S, \vee, \wedge)$ is called a *sublattice* of $\mathcal{L}$. A homomorphism between lattices is said to be a *lattice-homomorphism*. Similarly, an isomorphism between lattices is a *lattice-isomorphism*.

If $(L, \vee, \wedge)$ is a lattice, then an element $a \in L$ is *meet-irreducible* if $a = b \wedge c$ implies $a = b$ or $a = c$ for all $b, c \in L$. A We denote the set of all meet-irreducible elements $a \neq 1$ (in case L has a unit) of L by $\mathcal{M}(L)$. A *join-irreducible element* and the set $\mathcal{J}(L)$ are defined dually. The sets $\mathcal{M}(L)$ and $\mathcal{J}(L)$ inherit L's order relation, and will be regarded as an ordered set.

Lemma 2.13 *Let L be a lattice satisfying the* ACC.
(a) *If $a, b \in L$ and $b \nleq a$, then there exists $x \in \mathcal{M}(L)$ such that $a \leq x$ and $b \nleq x$.*
(b) $a = \bigwedge\{b \in \mathcal{M}(L) \mid a \leq b\}$.

Proof. (a) Suppose $b \nleq a$ and let us denote $S = \{x \in L \mid a \leq x \text{ and } b \nleq x\}$. The set S is nonempty since it contains a. Because L satisfies the ACC, there exists a maximal element x in S. We claim that x is in $\mathcal{M}(L)$. Suppose that $x = c \wedge d$ with $x < c$ and $x < d$. By the maximality of x, neither c nor d is in S. We have $a \leq x < c$, so $a \leq c$, and similarly, $a \leq d$. Therefore $c, d \notin S$ implies $b \leq c$ and $b \leq d$. But then $b \leq c \wedge d = x$, a contradiction! Thus, x is meet-irreducible and obviously $x \neq 1$ whenever L has a unit.

(b) Consider any $a \in L$. Let $T = \{x \in \mathcal{M}(L) \mid a \leq x\}$. Clearly, a is a lower bound for T. Let c be any lower bound for T. We claim that $c \leq a$. Suppose that $c \nleq a$. Then $a < a \vee c$ and hence $a \vee c \nleq a$. By (a) there exists $x \in T$ with $a \vee c \nleq x$. But $x \in T$ implies by the definition of T that $a \leq x$, and $c \leq x$ since c is a lower bound of T. Thus, x is an upper bound of $\{a, c\}$, and consequently $a \vee c \leq x$, a contradiction! Hence $c \leq a$ holds, which implies $a = \bigwedge T$. □

2.7 Join-semilattices

We have seen that a lattice can be defined as an algebra as well as an ordered set. Next we show that there is a similar relationship in the case of join-semilattices. We have the following two definitions.

- A nonempty ordered set $(S, \leq)$ is called a *join-semilattice*, if for all $a, b \in S$, the join $a \vee b$ exists.
- A *semilattice* is an algebra $(P, \circ)$ of type (2), where $\circ$ is an associative, commutative and idempotent operation.

These two notions are related as follows. If the algebra $(P, \circ)$ is a semilattice, then the condition $a \leq b$ if and only if $a \circ b = b$ defines a partial order $\leq$ on P such that $(P, \leq)$ is a $\vee$-semilattice and $a \vee b = a \circ b$. Similarly, if $(P, \leq)$ is a meet-semilattice, then the algebra $(P, \vee)$ is a semilattice in the sense of the second definition.

Proposition 2.14 *Suppose $(S, \leq)$ and $(P, \leq)$ are ordered sets and φ is a function from S to P.*

(a) *If $(S, \leq)$ and $(P, \leq)$ are meet-semilattices, then the following are equivalent.*
(1) *φ is an order-isomorphism.*
(2) *φ is an isomorphism $(S, \vee) \to (P, \vee)$.*

(b) *If $(S, \leq)$ and $(P, \leq)$ are lattices, then the following are equivalent.*
(1) *φ is an order-isomorphism.*
(2) *φ is a lattice-isomorphism $(S, \vee, \wedge) \to (P, \vee, \wedge)$.*

Proof. We show that (a) holds. Claim (b) can be proved similarly. It is obvious that in both cases (1) and (2) the function φ is a bijection. Suppose (1) holds, that is,

φ is an order-isomorphism. Because $a, b \leq a \vee b$, we have $\varphi(a) \vee \varphi(b) \leq \varphi(a \vee b)$. If $\varphi(a) \vee \varphi(b) \leq u$, then $\varphi(a) \leq u$ and $\varphi(b) \leq u$, which implies $a \leq \varphi^{-1}(u)$ and $b \leq \varphi^{-1}(u)$. Hence, $a \vee b \leq \varphi^{-1}(u)$ from which we get $\varphi(a \vee b) \leq u$. If we set $u = \varphi(a) \vee \varphi(b)$, then $\varphi(a) \vee \varphi(b) \geq \varphi(a \vee b)$. This implies $\varphi(a) \vee \varphi(b) = \varphi(a \vee b)$.

Suppose φ is an isomorphism from $(S, \vee)$ to $(P, \vee)$. If $a \leq b$ holds is S, then $\varphi(a) \vee \varphi(b) = \varphi(a \vee b) = \varphi(b)$, i.e., $\varphi(a) \leq \varphi(b)$. Conversely, if $\varphi(a) \leq \varphi(b)$, then $\varphi(b) = \varphi(a) \vee \varphi(b) = \varphi(a \vee b)$. Because φ is an injection this implies $b = a \vee b$, that is, $a \leq b$. □

By previous proposition it is obvious that if $(S, \vee, \wedge)$ and $(P, \vee, \wedge)$ are lattices and $\varphi : S \to P$ is a bijection, then φ is a lattice-isomorphism whenever φ is a homomorphism from $(S, \vee)$ to $(P, \vee)$.

Chapter 3

Knowledge bases

3.1 Knowledge bases and indiscernibility

We simply assume here that knowledge is an ability to partition objects, and by an object we mean anything which can be spoken of in the subject position of a natural language sentence. Objects need not to be atomic or indivisible. For mathematical reasons we often use equivalence relations instead of partitions, since there is bijective relationship between equivalences and partitions, and equivalences are easier to deal with. Hence, knowledge can be understood as a set of equivalence relations on a fixed universe. We need some formal definitions which are given below.

Let U be a nonempty set and $E(\subseteq \mathrm{Eq}(U))$ be a set of equivalence relations on U. Then the pair $\mathcal{K} = (U, E)$ is called a *knowledge base* and the set U is the *universe* of $\mathcal{K}$. Note that in [21] the set U is assumed to be finite, which implies that the set $E(\subseteq \mathrm{Eq}(U))$ is also finite. However, we do not make any general assumption about the cardinalities of U and E.

Example 3.1 This example is modified from an example appearing in [21].

Suppose we are given the set $U = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$ of toy blocks. Assume these toys have different colors (*red*, *blue*, *yellow*), shapes (*square*, *round*, *triangular*), and size (*small*, *large*). For example, a toy block can be *red*, *round*, and *small*. Hence, the set of toy blocks U can be classified according to color, shape, and size, for example, as follows:

x_1, x_3, x_7	are *red*,
x_2, x_4	are *blue*,
x_5, x_6, x_8	are *yellow*,
x_1, x_5	are *round*,
x_2, x_6	are *square*,
x_3, x_4, x_7, x_8	are *triangular*,
x_2, x_7, x_8	are *large*, and
x_1, x_3, x_4, x_5, x_6	are *small*.

These classifications can be considered as the equivalence relations e_1, e_2, and e_3 such that

$$\begin{aligned} U/e_1 &= \{\{x_1, x_3, x_7\}, \{x_2, x_4\}, \{x_5, x_6, x_8\}\}, \\ U/e_2 &= \{\{x_1, x_5\}, \{x_2, x_6\}, \{x_3, x_4, x_7, x_8\}\}, \quad \text{and} \\ U/e_3 &= \{\{x_2, x_7, x_8\}, \{x_1, x_3, x_4, x_5, x_6\}\}. \end{aligned}$$

If we denote $E = \{e_1, e_2, e_3\}$, then the pair $\mathcal{K} = (U, E)$ is a knowledge base. For all $e \in E$, each equivalence class of U/e consists of objects which are indiscernible with respect to knowledge e.

Because the intersection of equivalences is again an equivalence, we can form new classifications by applying this operation. For example, the sets

$$\begin{aligned} &\{x_1, x_3, x_7\} \cap \{x_3, x_4, x_7, x_8\} = \{x_3, x_7\}, \\ &\{x_2, x_4\} \cap \{x_2, x_6\} = \{x_2\}, \quad \text{and} \\ &\{x_5, x_6, x_8\} \cap \{x_3, x_4, x_7, x_8\} = \{x_8\}. \end{aligned}$$

are equivalence classes of $e_1 \cap e_2$ representing the combinations of *red* and *triangular*, *blue* and *square*, and *yellow* and *triangular*, respectively. Note that some combinations do not appear in this knowledge base. For example,

$$\begin{aligned} &\{x_2, x_4\} \cap \{x_1, x_5\} = \emptyset, \quad \text{and} \\ &\{x_1, x_3, x_7\} \cap \{x_2, x_6\} = \emptyset, \end{aligned}$$

which means that there are no *blue* and *round*, or *red* and *square* toy blocks.

By the previous example, we may derive new knowledge about objects by applying the set-theoretical operation of intersection. Assume $\mathcal{K} = (U, E)$ is a knowledge base. If $D \subseteq E$, then then D determines an equivalence $Ind(D) = \bigcap D$ on U, called the *indiscernibility relation* of D. The equivalence $Ind(D)$ represents the conjunction of knowledge presented by the individual equivalences in D. More precisely, if x and y are objects, then $(x, y) \in Ind(D)$ if and only if x and y are indiscernible with respect to all $e \in D$. In other words, two objects are discernible with respect to knowledge D if and only if there exists at least one $e \in D$ such that these objects are discernible with respect to knowledge e.

Let $\mathcal{K} = (U, E)$ be a knowledge base and assume D and F are subsets of E. The sets D and F are *equivalent*, denoted by $D \equiv F$, if $Ind(D) = Ind(F)$. Thus, if $D \equiv F$, then D and F define the same partition of objects. If $Ind(F) \subseteq Ind(D)$, then the knowledge D is said to be *dependent on* the knowledge F in $\mathcal{K}$, denoted by $F \to D$ $(\mathcal{K})$. Sometimes we write simply $F \to D$ if there is no danger of confusion. If $F \to D$, then the combined knowledge represented by D is derivable from the combined knowledge represented by F, that is, if two objects are indiscernible with respect to the knowledge F and $F \to D$ holds, then these two objects are also indiscernible with respect to the knowledge D.

In the following we present some general facts concerning the concepts defined above. Statements (a)–(d) follow directly from the definition of the operator Ind, and also condition (e) is obvious.

Lemma 3.2 *If $\mathcal{D} = (U, E)$ is a knowledge base and $D, F \subseteq E$, then the following facts hold.*

(a) $Ind(\emptyset) = U^2$.
(b) $Ind(D \cup F) = Ind(D) \cap Ind(F)$.
(c) $D \subseteq F$ *implies* $Ind(F) \subseteq Ind(D)$.
(d) $Ind(D) \subseteq e$ *for all* $e \in D$.
(e) $D \equiv F$ *if and only if* $D \to F$ *and* $F \to D$. □

Equation (a) of Lemma 3.2 says that if we have no knowledge, that is, our set of equivalences is empty, we cannot discern any objects. Statement (b) shows how the indiscernibility relation of the union of sets depends on the indiscernibility relation of the components of the union. Namely, two objects are indiscernible with respect to $D \cup F$ if and only if they are indiscernible with respect to D and F. By (c) any set is dependent on its supersets. Statement (d) is actually a useful special case of (c). By (e) subsets are equivalent if and only if they are dependent on each other.

3.2 Indispensable elements, independent sets, and reducts

The fundamental problem of this section is in [21] referred to as that of *knowledge reduction*. Here the central role is played by the concepts of indispensable elements, independent subsets, and the core and reducts of knowledge. Recall that we have omitted the finiteness of the sets U, which will slightly complicate our considerations.

Let $\mathcal{K} = (U, E)$ be a knowledge base and $D \subseteq E$. We say that an equivalence $e \in D$ is *indispensable* in D if $Ind(D) \neq Ind(D - \{e\})$ (i.e. $Ind(D) \subset Ind(D - \{e\})$). If $e \in D$ is indispensable in D, then the combined knowledge given by D is not equivalent to the combined knowledge given by $D - \{e\}$, that is, there are at least two objects which are indiscernible with knowledge $D - \{e\}$, but discernible with respect to knowledge D.

A subset $D(\subseteq E)$ is *independent* if all $e \in D$ are indispensable in D; otherwise D is *dependent*. It is clear that D is independent if and only if D is not is not equivalent to any proper subset of D.

Lemma 3.3 *Suppose* $\mathcal{K} = (U, E)$ *is a knowledge base and let* $D \subseteq F \subseteq E$.
(a) *An element* $e \in D$ *is indispensable in* D *if and only if* $Ind(D - \{e\}) \not\subseteq e$.
(b) *If* e *is indispensable in* F, *then* e *is indispensable in* D.
(c) *If* D *is independent, then* D *is an antichain with respect to the inclusion relation.*

Proof. (a) By Lemma 3.2(b), $Ind(D) = Ind(D - \{e\}) \cap e$ for all $e \in D$, which implies that for all $e \in D$, the condition $Ind(D - \{e\}) \subseteq e$ is equivalent to $Ind(D) = Ind(D - \{e\})$.

(b) Suppose e is indispensable in F and $D \subseteq F$. Then the conditions $Ind(F) \not\subseteq e$ and $Ind(F) \subseteq Ind(D)$ imply $Ind(D) \not\subseteq e$, that is, e is indispensable in D.

(c) We verify the contrapositive of the claim, i.e., if D is not an antichain, then D is dependent. Assume that there are two distinct equivalences e and f in D which satisfy $e \subseteq f$. Then by (a), the relation f is not indispensable in $\{e, f\}$. Because $\{e, f\} \subseteq D$, f is not indispensable on D by (b), which implies that D is dependent. □

Statement (a) of Lemma 3.3 is useful for deciding whether an element is indispensable in a subset. From (b) it follows that every superset of a dependent set is dependent, and all subsets of an independent set are independent. By (c) it is clear that if $e \subseteq f$ for some distinct $e, f \in D$, then D is dependent. Further, if D is a chain, then all independent subsets of D are of the form $\{e\}$, where $e \in D$.

Assume $\mathcal{K} = (U, E)$ is a knowledge base and $D \subseteq E$. The set of all indispensable elements of D will be called the *core* of D, and will be denoted by $CORE_{\mathcal{K}}(D)$. A subset F of D is said to be a *reduct* of D if $Ind(D) = Ind(F)$ and F is independent. The set of all reducts of D is denoted by $RED_{\mathcal{K}}(D)$. Obviously, a reduct of D is a minimal subset of D which represents the same knowledge as D itself.

Suppose $\mathcal{K} = (U, E)$ is a knowledge base. Let

$$\mathcal{F} = \{Ind(F) \mid F \text{ is a finite subset of } E\}.$$

We may write the following lemma.

Lemma 3.4 *Assume $\mathcal{K} = (U, E)$ is a knowledge base. If $\mathcal{F}$ satisfies the* DCC, *then for all nonempty subsets D of E there exists a finite subset F of D which satisfies $Ind(D) = Ind(F)$.*

Proof. Assume $\mathcal{F}$ satisfies the DCC and let D be a subset of E. Let us denote

$$\mathcal{F}(D) = \{Ind(F) \mid F \text{ is a finite subset of } D\}.$$

Because $\emptyset \in \mathcal{F}(D)$, $\mathcal{F}(D)$ is a nonempty subset of $\mathcal{F}$. By assumption $\mathcal{F}$ satisfies the DCC, which implies by the dual of Lemma 2.4 that $\mathcal{F}(D)$ has a minimal element $Ind(F)$ for some finite $F \subseteq D$. For all $e \in D$, $Ind(F \cup \{e\}) \in \mathcal{F}(D)$ and trivially $Ind(F \cup \{e\}) \subseteq Ind(F)$. Because $Ind(F)$ is minimal, this implies $Ind(F) = Ind(F \cup \{e\}) = Ind(F) \cap e$ for all $e \in D$. Hence, $Ind(F) \subseteq e$ for all $e \in D$, and $Ind(F) \subseteq \bigcap D = Ind(D)$ holds. On the other hand, $F \subseteq D$ implies $Ind(D) \subseteq Ind(F)$. Thus, $Ind(D) = Ind(F)$. □

Proposition 3.5 *Assume $\mathcal{K} = (U, E)$ is a knowledge base and $\mathcal{F} = \{Ind(F) \mid F$ is a finite subset of $E\}$.*

(a) *If $\mathcal{F}$ satisfies the* DCC, *then every subset of E has a finite reduct.*

(b) *If $\mathcal{F}$ satisfies the* DCC, *then $CORE_{\mathcal{K}}(D) = \bigcap RED_{\mathcal{K}}(D)$ for all $D \subseteq E$.*

(c) *If E has no minimal element with respect to the inclusion relation, then there exists a subset $D(\subseteq E)$ which has no reducts; moreover $CORE_{\mathcal{K}}(D) \neq \bigcap RED_{\mathcal{K}}(D)$ holds.*

Proof. (a) Suppose $D \subseteq E$ and $\mathcal{F}$ satisfies the DCC. Then by Lemma 3.4 there exists a finite subset $F \subseteq D$ such that $Ind(D) = Ind(F)$. Suppose $F = \{e_1, \ldots, e_n\}$, $n \in \mathbb{N}_0$ and let us define inductively the following sets F_i for all i, $0 \leq i \leq n$.

$$F_0 = F \qquad \text{and} \qquad F_{i+1} = \begin{cases} F_i - \{e_{i+1}\} & \text{if } Ind(F_i - \{e_{i+1}\}) \subseteq e_{i+1}, \\ F_i & \text{otherwise.} \end{cases}$$

Obviously, $F_n \subseteq F_{n-1} \subseteq \cdots \subseteq F_1 \subseteq F_0 \subseteq D$ and $Ind(D) = Ind(F_0) = Ind(F_1) = \cdots = Ind(F_n)$. Assume F_n is dependent, that is, $Ind(F_n - \{e_{i+1}\}) \subseteq$

e_{i+1} for some i, $0 \leq i \leq n-1$. Because $F_n \subseteq F_i$ we have by Lemma 3.2(c) that $Ind(F_i - \{e_{i+1}\}) \subseteq Ind(F_n - \{e_{i+1}\}) \subseteq e_{i+1}$. This implies by the definition that $e_{i+1} \notin F_{i+1}$ and $e_{i+1} \notin F_n$, a contradiction! Hence, F_n is a reduct of D.

(b) Assume $e \in CORE_{\mathcal{K}}(D)$ and $e \notin F$ for some $F \in RED_{\mathcal{K}}(D)$. Now $F \subseteq D - \{e\} \subseteq D$, which implies $Ind(D) \subseteq Ind(D - \{e\}) \subseteq Ind(F)$. Because $Ind(D) = Ind(F)$, we have $Ind(D) = Ind(D - \{e\})$, that is, e is dispensable in D, a contradiction! Conversely, suppose $e \in \bigcap RED_{\mathcal{K}}(D)$ and $Ind(D - \{e\}) = Ind(D)$. Because $\mathcal{F}$ satisfies the DCC, there exists a reduct F of $D - \{e\}$. The equation $Ind(D - \{e\}) = Ind(D)$ implies that the set F is a reduct of D. Because $e \notin F$, this implies $e \notin \bigcap RED_{\mathcal{K}}(D)$, a contradiction!

(c) Suppose E has no minimal element. Then by the dual of Lemma 2.4 there exists an infinite descending chain $e_1 \supset e_2 \supset \cdots$ in E. Let us denote $D = \{e_i \mid i \in \mathbb{N}\}$. As we have noted, all independent subsets of the chain D are of the form $\{e_i\}$, where $i \in \mathbb{N}$. Assume $\{e_k\}$ is a reduct of D for some $k \in \mathbb{N}$. This implies $Ind(D) = Ind(\{e_k\}) = e_k$. But now there exists an element e_{k+1} in D which satisfies $Ind(D) \subseteq e_{k+1} \subset e_k$, a contradiction! Hence, D has no reducts. Moreover, $\bigcap RED_{\mathcal{K}}(D) = \{e \in E \mid e \text{ belongs to all reducts of } D\} = E$, and $Ind(D - \{e_i\}) \subseteq e_i$ for all $i \in \mathbb{N}$ which implies $CORE_{\mathcal{K}}(D) = \emptyset$. Hence, $CORE_{\mathcal{K}}(D) \neq \bigcap RED_{\mathcal{K}}(D)$. □

In Pawlak's original definition of knowledge bases the universe U is assumed to be finite. Because $\text{Eq}(U) \subseteq \wp(U \times U)$, the set $\text{Eq}(U)$ is finite whenever U is finite, which implies trivially that every subset $E(\subseteq \text{Eq}(U))$ is finite. Obviously, if $\mathcal{K} = (U, E)$ is a knowledge base such that E is finite, then the set $\mathcal{F} = \{Ind(F) \mid F \text{ is a finite subset of } E\}$ is finite, and it satisfies the DCC. Hence, for a knowledge base in which either of the sets U of E is finite, every subset has at least one finite reduct and $CORE_{\mathcal{K}}(D) = \bigcap RED_{\mathcal{K}}(D)$ for all subset D of the knowledge base $\mathcal{K}$

Example 3.6 Suppose $U = \mathbb{N}$. For each $i \in \mathbb{N}$, we define an equivalence e_i which equivalence classes are $U/e_i = \{\{1\}, \ldots, \{i\}, \{i+1, i+2, \ldots\}\}$. If we denote $E = \{e_i \mid i \in \mathbb{N}\}$, then the pair $\mathcal{K} = (U, E)$ is a knowledge base. Obviously, $e_j \subset e_i$ for any $i < j$, and all independent subsets of E are the sets $\{e_i\}$, where $i \in \mathbb{N}$. The set E has no minimal elements which fact obviously implies that E does not have any reducts. Moreover, $CORE_{\mathcal{K}}(E) = \emptyset$ and $\bigcap RED_{\mathcal{K}}(E) = E$, that is, the equation $CORE_{\mathcal{K}}(E) = \bigcap RED_{\mathcal{K}}(E)$ does not hold.

If F is a finite nonempty subset of E, that is, $F = \{e_{i_1}, \ldots, e_{i_n}\}$, where $i_1 < \cdots < i_n$ for some $n \in \mathbb{N}$, then $Ind(E) = e_{i_n}$. Thus, $\{e_{i_n}\}$ is the only reduct of F. Moreover, the equation $CORE_{\mathcal{K}}(F) = \bigcap RED_{\mathcal{K}}(F)$ holds.

3.3 Dependence in knowledge bases and dependence in universal algebra

In this section we study how the dependence defined in knowledge bases relates to the abstract dependence in universal algebra. The definition of abstract dependence can be found in [3, 5, 6], for example.

Let P be a set. An *abstract dependence* on P is a system $\boldsymbol{D}$ of subsets of P such that any subset S of P belongs to $\boldsymbol{D}$ if and only if there is a finite nonempty subset T of S with $T \in \boldsymbol{D}$. A subset $S \subseteq P$ is called *dependent* if $S \in \boldsymbol{D}$; otherwise it is called *independent*. By this definition, every subset of an independent set is independent; in particular, the empty set is independent. Equivalently, every superset of a dependent set is dependent.

Example 3.7 Let $P = \{1, 2, 3\}$ and let $\boldsymbol{D} = \{\{2\}, \{1, 2\}, \{2, 3\}, \{1, 2, 3\}\}$. Clearly, $\boldsymbol{D}$ is an abstract dependence on P.

As observed in [21], if $\mathcal{K} = (U, E)$ is a knowledge base in which U is finite, then the set of all dependent subsets of $\mathcal{K}$ is an abstract dependence. However, if the sets U and E are infinite, this fact does not necessarily hold. Next we shall present a condition which guarantees that the set of all dependent subsets of a knowledge base is an abstract dependence.

Proposition 3.8 *Suppose $\mathcal{K} = (U, E)$ is a knowledge base and let $\mathcal{F} = \{Ind(F) \mid F$ is a finite subset of $E\}$. If the set $\mathcal{F}$ satisfies the* DCC, *then the set of all dependent subsets of E is an abstract dependence on E.*

Proof. Assume $D \subseteq E$ is dependent and $\mathcal{F}$ satisfies the DCC. If D is finite, then trivially there is a finite dependent subset D of D. If D is infinite, then there exists $e \in D$ such that $Ind(D - \{e\}) \subseteq e$. Because $\mathcal{F}$ satisfies the DCC, there exists a finite subset F of $D - \{e\}$ such that $Ind(D - \{e\}) = Ind(F)$. The facts $e \notin F$ and $Ind(F) \subseteq e$ imply that the set $F \cup \{e\}$ is a finite dependent subset of D. □

The implication of Proposition 3.8 does not hold in other direction, as the following example shows.

Example 3.9 Let us consider the knowledge base $\mathcal{K} = (U, E)$ of Example 3.6. It is obvious that the set $\mathcal{F}$ does not satisfy the DCC. Suppose $D(\subseteq E)$ is dependent. Then D contains at least two elements, that is, there exists a subset $\{e_k, e_l\}$ of D. Now either $e_k \subseteq e_l$ or $e_l \subseteq e_k$ holds, i.e., $\{e_k, e_l\}$ is a finite dependent subset of D. Hence, the set of all dependent subsets of $\mathcal{K}$ is an abstract dependence on E.

Chapter 4

Information systems

4.1 Information systems and indiscernibility

In this chapter we study information systems. The notion of information systems is introduced by Pawlak in [19] and it is investigated by several authors (see e.g. [15, 16, 17, 22, 23, 24, 25, 26, 27, 28]). An *information system* is a triple $\mathcal{S} = (U, A, \{V_a\}_{a\in A})$, where U is a nonempty set of *objects*, A is a nonempty set of *attributes*, and $\{V_a\}_{a\in A}$ is an indexed set of *values of attributes*. Each attribute $a \in A$ is a function $a : U \to V_a$. Moreover, we denote $V = \bigcup_{a\in A} V_a$. Usually the sets U, A, and V are assumed to be finite, which is actually a very natural assumption. However, until further notice, we do not assume anything about the cardinalities of these sets.

For any $a \in A$ the kernel, $\ker a = \{(x, y) \in U \times U \mid a(x) = a(y)\}$, of the attribute a is now equivalence on U. We may consider that the relation $\ker a$ represents knowledge about objects in the sense that two objects x and y are in the relation $\ker a$ if they are indiscernible with respect to an attribute a, that is, they have the same value for the attribute a.

Example 4.1 An information system $\mathcal{S}$ in which the sets U, A, and V are finite can be represented by a table. The rows of the table are labeled by the objects, and the columns by the attributes of the system $\mathcal{S}$. In the intersection of the row labeled by an object x and the column labeled by an attribute a we find the value $a(x)$.

Let us consider a simple example of an information system which is taken from [24]. In the example $\mathcal{S} = (U, A, \{V_a\}_{a\in A})$, where $U = \{x_1, \ldots, x_4\}$, $A = \{1, \ldots, 4\}$, $V_1 = V_2 = V_3 = \{0, 1\}$, $V_4 = \{0, 1, 2\}$, and the values of the attributes are defined as in Table 1.

	1	2	3	4
x_1	0	0	0	0
x_2	0	1	0	2
x_3	1	1	0	1
x_4	0	1	1	2

Table 1

For example, the objects x_1 and x_2 are indiscernible with respect to attributes 1 and 3.

Any information system $\mathcal{S} = (U, A, \{V_a\}_{a\in A})$ defines a knowledge base as follows. If we set $E_{\mathcal{S}} = \{\ker a \mid a \in A\}$, then the system $\mathcal{K}_{\mathcal{S}} = (U, E_{\mathcal{S}})$ is a knowledge base. However, we must note that $a_1 \neq a_2$ does not necessarily imply $\ker a_1 \neq \ker a_2$, that is, two distinct attributes may define the same equivalence on the set of objects. Similarly, for every knowledge base $\mathcal{K}$ there exists an information system $\mathcal{S}_{\mathcal{K}}$ such that $\mathcal{K} = \mathcal{K}_{(\mathcal{S}_{\mathcal{K}})}$. Next, we shall present this construction. Suppose $\mathcal{K} = (U, E)$ is a knowledge base. Let us set

- $A_{\mathcal{K}} = \{v_e \mid e \in E\}$, where v_e denotes the canonical map $U \to U/e$, $x \mapsto x/e$, of the equivalence e, and
- $V_{v_e} = U/e$ for all $e \in E$.

Clearly, the system $\mathcal{S}_{\mathcal{K}} = (U, A_{\mathcal{K}}, \{V_a\}_{a\in A_{\mathcal{K}}})$ is an information system such that $\mathcal{K} = \mathcal{K}_{(\mathcal{S}_{\mathcal{K}})}$. Note, that $\mathcal{S} = \mathcal{S}_{(\mathcal{K}_{\mathcal{S}})}$ does not usually hold.

Let $\mathcal{S} = (U, A, \{V_a\}_{a\in A})$ be an information system. For all subsets $B (\subseteq A)$ of attributes we define the following relation

$$Ind(B) = \bigcap_{a\in B} \ker a. \tag{4.1}$$

The relation $Ind(B)$ is called the *indiscernibility relation of the subset of attributes* B. If $(x, y) \in Ind(B)$, then objects x and y are said to be B*-indiscernible*. Hence, x and y are B-indiscernible whenever they are indiscernible with respect to all attributes in B. Because for all $a \in A$, the relation $\ker a$ is an equivalence and the intersection of equivalence relations is again an equivalence, the relation $Ind(B)$ is also an equivalence. The partition of the objects corresponding to the equivalence relation $Ind(B)$ can be viewed as a classification of objects, in which the equivalence classes of $Ind(B)$ consist of objects which are B-indiscernible. Note that the relation $Ind(B)$ defined in $\mathcal{S}$ equals to the relation $Ind(\{\ker a \mid a \in B\})$ which is an indiscernibility relation defined in the knowledge base $\mathcal{K}_{\mathcal{S}}$.

Next we consider Ind as a function from $\wp(A)$ to $\mathrm{Eq}(U)$, that is, the function Ind assigns to each subset of attributes the corresponding indiscerniblitity relation. The following facts are obvious.

$$Ind(\emptyset) = U \times U \tag{4.2}$$

and

$$\text{If } B \subseteq C, \text{ then } Ind(C) \subseteq Ind(B). \tag{4.3}$$

Intuitively, we cannot discern objects by means of the empty set, and if two objects are indiscernible with respect to a set C of attributes, they certainly are indiscernible with respect to any subset B of C. By (4.3), the function Ind is order-reversing.

Lemma 4.2 *If $\mathcal{S} = (U, A, \{V_a\}_{a\in A})$ is an information system and $\{B_i \mid i \in I\}$ is a family of subsets of A, then*

(a) $\bigcap_{i\in I} Ind(B_i) = Ind(\bigcup_{i\in I} B_i)$, *and*

(b) $\bigcup_{i\in I} Ind(B_i) \subseteq Ind(\bigcap_{i\in I} B_i)$.

Proof. Suppose $\{B_i \mid i \in I\}$ is a family of subsets of A.

(a) If $I = \emptyset$, then $\bigcap_{i\in\emptyset} Ind(B_i) = \bigcap \emptyset = U \times U = Ind(\emptyset) = Ind(\bigcup_{i\in\emptyset} B_i)$. If $I \neq \emptyset$, then $(x, y) \in \bigcap_{i\in I} Ind(B_i) \Leftrightarrow (x, y) \in Ind(B_i)$ for all $i \in I \Leftrightarrow$ for all $i \in I$, $x, y \in \ker a$ for all $a \in B_i \Leftrightarrow (x, y) \in \ker a$ for all $a \in \bigcup_{i\in I} B_i \Leftrightarrow (x, y) \in Ind(\bigcup_{i\in I} B_i)$.

(b) If $I = \emptyset$, then $\bigcup_{i\in\emptyset} Ind(B_i) = \bigcup \emptyset = \emptyset \subseteq Ind(A) = Ind(\bigcap_{i\in\emptyset} B_i)$. If $I \neq \emptyset$, then obviously $\bigcap_{i\in I} B_i \subseteq B_i$ for all $i \in I$. By (4.3), $Ind(B_i) \subseteq Ind(\bigcap_{i\in I} B_i)$ for all $i \in I$. Hence, $\bigcup_{i\in I} Ind(B_i) \subseteq Ind(\bigcap_{i\in I} B_i)$. □

Note that the equation $\bigcup_{i\in I} Ind(B_i) = Ind(\bigcap_{i\in I} B_i)$ does not usually hold.

4.2 The complete lattice of indiscernibility relations

In this section we study the structure of the set of all indiscernibility relations in an information system. The results given here are novel in a sense that in our considerations the sets U, A, and V are allowed to be infinite.

By Lemma 4.2(a) and (4.3), the function $Ind : \wp(A) \to \mathrm{Eq}(U)$ is a homomorphism from the semilattice $(\wp(A), \cup)$ to the semilattice $(\mathrm{Eq}(U), \cap)$. Hence by Lemma 2.8, $(\{Ind(B) \mid B \subseteq A\}, \cap)$ is a subalgebra of $(\mathrm{Eq}(U), \cap)$. Further, the function Ind is an epimorphism $(\wp(A), \cup) \to (\{Ind(B) \mid B \subseteq A\}, \cap)$.

By the Homomorphism Theorem we get the following result.

Proposition 4.3 *Let $\mathcal{S} = (U, A, \{V_a\}_{a\in A})$ be an information system. The semilattices $(\wp(A)/K_\mathcal{S}, \vee)$ and $(\{Ind(B) \mid B \subseteq A\}, \cap)$ are isomorphic, where $K_\mathcal{S} = \ker Ind$. The operation $\vee$ is defined in $\wp(A)$ by*

$$B/K_\mathcal{S} \vee C/K_\mathcal{S} = (B \cup C)/K_\mathcal{S}.$$

The isomorphism is $\varphi : B/K_\mathcal{S} \mapsto Ind(B)$. □

The situation of the previous proposition is illustrated by Figure 2.

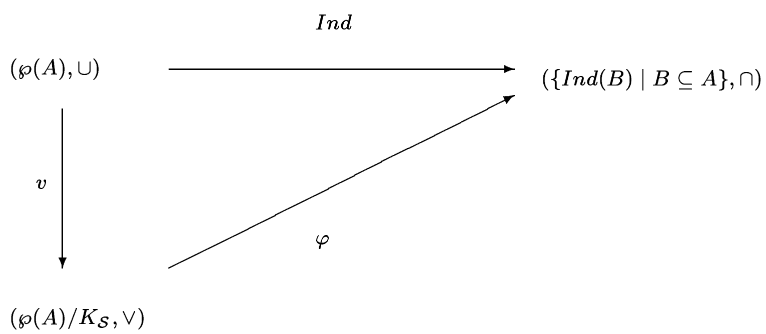

Figure 2

The join-semilattice corresponding to the quotient semilattice $(\wp(A)/K_\mathcal{S}, \vee)$ is an ordered set $(\wp(A)/K_\mathcal{S}, \leq)$ in which the partial order is given by the condition

(4.4) $$B/K_\mathcal{S} \leq C/K_\mathcal{S} \text{ if and only if } (B \cup C)/K_\mathcal{S} = C/K_\mathcal{S}.$$

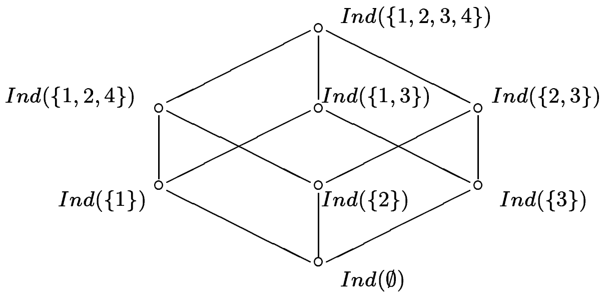

Figure 3

Note that the join-semilattice corresponding to the semilattice $(\{Ind(B) \mid B \subseteq A\}, \cap)$ is $(\{Ind(B) \mid B \subseteq A\}, \supseteq)$.

Example 4.4 Let us consider the information system $\mathcal{S}$ of Example 4.1. It can be easily verified that the equivalence classes of the equivalences $Ind(B)$, $B \subseteq A$, are the following:

- $U/Ind(\emptyset) = \{U\}$;
- $U/Ind(\{1\}) = \{\{x_1, x_2, x_4\}, \{x_3\}\}$;
- $U/Ind(\{2\}) = \{\{x_1\}, \{x_2, x_3, x_4\}\}$;
- $U/Ind(\{3\}) = \{\{x_1, x_2, x_3\}, \{x_4\}\}$;
- $U/Ind(\{4\}) = U/Ind(\{1,2\}) = U/Ind(\{1,4\}) = U/Ind(\{2,4\}) = U/Ind(\{1,2,4\})$
  $= \{\{x_1\}, \{x_2, x_4\}, \{x_3\}\}$;
- $U/Ind(\{1,3\}) = \{\{x_1, x_2\}, \{x_3\}, \{x_4\}\}$;
- $U/Ind(\{2,3\}) = \{\{x_1\}, \{x_2, x_3\}, \{x_4\}\}$;
- $U/Ind(\{3,4\}) = U/Ind(\{1,2,3\}) = U/Ind(\{1,3,4\}) = U/Ind(\{2,3,4\})$
  $= U/Ind(\{1,2,3,4\}) = \{\{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}\}$.

The congruence classes of $K_{\mathcal{S}}$ are $\{\emptyset\}$, $\{\{1\}\}$, $\{\{2\}\}$, $\{\{3\}\}$, $\{\{4\}, \{1,2\}, \{1,4\}, \{2,4\}, \{1,2,4\}\}$, $\{\{1,3\}\}$, $\{\{2,3\}\}$, and $\{\{3,4\}, \{1,2,3\}, \{1,3,4\}, \{2,3,4\}, \{1,2,3,4\}\}$. The join-semilattice $(\{Ind(B) \mid B \subseteq A\}, \supseteq)$ is presented in Figure 3.

Lemma 4.5 *Suppose $\mathcal{S} = (U, A, \{V_a\}_{a \in A})$ is an information system and $B \subseteq A$. Then $\bigcup B/K_{\mathcal{S}} \in B/K_{\mathcal{S}}$, and $\bigcup B/K_{\mathcal{S}}$ is the greatest element in the congruence class $B/K_{\mathcal{S}}$.*

Proof. Suppose $B \subseteq A$. Then $Ind(\bigcup B/K_{\mathcal{S}}) = \bigcap_{C \in B/K_{\mathcal{S}}} Ind(C) = Ind(B)$. It is obvious that $C \subseteq \bigcup B/K_{\mathcal{S}}$ for all $C \in B/K_{\mathcal{S}}$. □

Let us now define a function

$$\begin{aligned} \mathcal{C}_{\mathcal{S}} : \wp(A) &\to \wp(A), \\ B &\mapsto \bigcup B/K_{\mathcal{S}}. \end{aligned}$$

Proposition 4.6 *If $\mathcal{S} = (U, A, \{V_a\}_{a\in A}$ is an information system, then the following facts hold.*

(a) $\mathcal{C}_{\mathcal{S}}(B) = \{a \in A \mid Ind(B) \subseteq Ind(\{a\})\}$ *for all* $B \subseteq A$.

(b) $\mathcal{C}_{\mathcal{S}}$ *is a closure operator.*

(c) $\ker \mathcal{C}_{\mathcal{S}} = K_{\mathcal{S}}$.

Proof. (a) If $a \in \mathcal{C}_{\mathcal{S}}(B)$, then there exists a $C \in B/K_{\mathcal{S}}$ such that $a \in C$. Hence, $Ind(B) = Ind(C) \subseteq Ind(\{a\})$. Conversely, if $Ind(B) \subseteq Ind(\{a\})$, then $Ind(B \cup \{a\}) = Ind(B) \cap Ind(\{a\}) = Ind(B)$, that is, $B \cup \{a\} \in B/K_{\mathcal{S}}$.

(b) We shall show that $\mathcal{C}_{\mathcal{S}}$ is (i) extensive, (ii) isotone, and (iii) idempotent. Suppose $B \subseteq A$. (i) Because for all $a \in B$, $Ind(B) \subseteq Ind(\{a\})$, we get $B \subseteq \mathcal{C}_{\mathcal{S}}(B)$. (ii) If $B \subseteq C$ and $a \in \mathcal{C}_{\mathcal{S}}(B)$, then $Ind(C) \subseteq Ind(B) \subseteq Ind(\{a\})$, that is, $a \in \mathcal{C}_{\mathcal{S}}(C)$. (iii) By (i) and (ii) it is clear that $\mathcal{C}_{\mathcal{S}}(B) \subseteq \mathcal{C}_{\mathcal{S}}(\mathcal{C}_{\mathcal{S}}(B))$. Suppose $a \in \mathcal{C}_{\mathcal{S}}(\mathcal{C}_{\mathcal{S}}(B))$. Then $Ind(\mathcal{C}_{\mathcal{S}}(B)) \subseteq IND(\{a\})$. Because $Ind(\mathcal{C}_{\mathcal{S}}(B)) = Ind(B)$, this implies $a \in \mathcal{C}_{\mathcal{S}}(B)$.

(c) Let $B, C \subseteq A$. If $(B, C) \in \ker \mathcal{C}_{\mathcal{S}}$, then $\mathcal{C}_{\mathcal{S}}(B) = \mathcal{C}_{\mathcal{S}}(C)$. Now $Ind(B) = Ind(\mathcal{C}_{\mathcal{S}}(B)) = Ind(\mathcal{C}_{\mathcal{S}}(B)) = Ind(C)$, that is, $(B, C) \in K_{\mathcal{S}}$. If $(B, C) \in K_{\mathcal{S}}$, then $\mathcal{C}_{\mathcal{S}}(B) = \bigcup B/K_{\mathcal{S}} = \bigcup C/K_{\mathcal{S}} = \mathcal{C}_{\mathcal{S}}(C)$. □

For an information system $\mathcal{S}$ we denote by $\mathcal{L}_{\mathcal{S}}$ the closure system on A corresponding to the closure operator $\mathcal{C}_{\mathcal{S}}$. The ordered set $(\mathcal{L}_{\mathcal{S}}, \subseteq)$ is a complete lattice in which

$$\begin{aligned} \bigwedge\{C_i \mid i \in I\} &= \bigcap_{i\in I} C_i\ ; \\ \bigvee\{C_i \mid i \in I\} &= \mathcal{C}_{\mathcal{S}}(\bigcup_{i\in I} C_i). \end{aligned}$$

By Lemma 2.6(a), for all $B, C \in \wp(A)$,

$$\mathcal{C}_{\mathcal{S}}(B \cup C) = \mathcal{C}_{\mathcal{S}}(\mathcal{C}_{\mathcal{S}}(B) \cup \mathcal{C}_{\mathcal{S}}(C)) = \mathcal{C}_{\mathcal{S}}(B) \vee_{\mathcal{L}_{\mathcal{S}}} \mathcal{C}_{\mathcal{S}}(B).$$

Thus, the function $\mathcal{C}_{\mathcal{S}}$ is a homomorphism from $(\wp(A), \cup)$ onto $(\mathcal{L}_{\mathcal{S}}, \vee_{\mathcal{C}_{\mathcal{S}}})$. We can write the following proposition.

Proposition 4.7 *If $\mathcal{S} = (U, A, \{V_a\}_{a\in A})$ is an information system, then*

(a) $(\{Ind(B) \mid B \subseteq A\}, \cap) \cong (\wp(A)/K_{\mathcal{S}}, \vee) \cong (\mathcal{L}_{\mathcal{S}}, \vee_{\mathcal{L}_{\mathcal{S}}})$, *and*

(b) $(\{Ind(B) \mid B \subseteq A\}, \supseteq) \cong (\wp(A)/K_{\mathcal{S}}, \leq) \cong (\mathcal{L}_{\mathcal{S}}, \subseteq)$.

Proof. (a) As we have seen in Proposition 4.3, the semilattices $(Ind(B) \mid B \subseteq A\}, \cap)$ and $(\wp(A)/K_{\mathcal{S}}, \vee)$ and isomorphic. The isomorphism of semilattices $(\wp(A)/K_{\mathcal{S}}, \vee)$ and $(\mathcal{L}_{\mathcal{S}}, \vee_{\mathcal{L}_{\mathcal{S}}})$ is clear by the Homomorphism Theorem and Proposition 4.6(c). Statement (b) is obvious by Proposition 2.14(a). □

Theorem 4.8 *Let $\mathcal{S} = (U, A, \{V_a\}_{a \in A})$ be an information system.*

(a) $(\{Ind(B) \mid B \subseteq A\}, \subseteq)$ *is a complete lattice in which for all* $\{B_i \mid i \in I\} \subseteq \wp(A)$,

$$\begin{aligned}\bigwedge\{Ind(B_i) \mid i \in I\} &= \bigcap_{i \in I} Ind(B_i) = Ind(\bigcup_{i \in I} B_i);\\ \bigvee\{Ind(B_i) \mid i \in I\} &= Ind(\bigcap_{i \in I} \mathcal{C}_{\mathcal{S}}(B_i)).\end{aligned}$$

(b) $(\wp(A)/K_{\mathcal{S}}, \leq)$ *is a complete lattice in which for all* $\{B_i \mid i \in I\} \subseteq \wp(A)$,

$$\begin{aligned}\bigwedge\{B_i/K_{\mathcal{S}} \mid i \in I\} &= (\bigcap_{i \in I} \mathcal{C}_{\mathcal{S}}(B_i))/K_{\mathcal{S}};\\ \bigvee\{B_i/K_{\mathcal{S}} \mid i \in I\} &= (\bigcup_{i \in I} B_i)/K_{\mathcal{S}}.\end{aligned}$$

Proof. (a) By (4.2) and (4.3), the set $\{Ind(B) \mid B \subseteq A\}$ has the top element $\top = Ind(\emptyset)$ and the bottom element $\bot = Ind(A)$. If $\{B_i \mid i \in I\}$ is a subset of $\wp(A)$, then it quite obvious by Lemma 4.2(a) that $\bigwedge\{Ind(B_i) \mid i \in I\} = \bigcap_{i \in I} Ind(B_i) = Ind(\bigcup_{i \in I} B_i)$.

For the rest, we show first that

$$\bigcup\{C \subseteq A \mid Ind(B_i) \subseteq Ind(C) \text{ for all } i \in I\} = \bigcap_{i \in I} \mathcal{C}_{\mathcal{S}}(B_i).$$

Suppose $a \in \bigcup\{C \subseteq A \mid Ind(B_i) \subseteq Ind(C) \text{ for all } i \in I\}$. Hence, there exists C such that $a \in C$ and $Ind(B_i) \subseteq Ind(C)$ for all $i \in I$. Because $\{a\} \subseteq C$, this implies $Ind(B_i) \subseteq Ind(C) \subseteq Ind(\{a\})$ for all $i \in I$. Then $a \in \mathcal{C}_{\mathcal{S}}(B_i)$ for all $i \in I$, i.e., $a \in \bigcap_{i \in I} \mathcal{C}_{\mathcal{S}}(B_i)$. Conversely, if $a \in \bigcap_{i \in I} \mathcal{C}_{\mathcal{S}}(B_i)$, then $Ind(B_i) \subseteq Ind(\{a\})$ for all $i \in I$. Hence, $\{a\} \in \{C \subseteq A \mid Ind(B_i) \subseteq Ind(C) \text{ for all } i \in I\}$, that is, $a \in \bigcup\{C \subseteq A \mid Ind(B_i) \subseteq Ind(C) \text{ for all } i \in I\}$. By Lemma 2.1,

$$\begin{aligned}\bigvee\{Ind(B_i) \mid i \in I\} &= \bigwedge\{Ind(B_i) \mid i \in I\}^u\\ &= \bigcap\{Ind(C) \mid Ind(B_i) \subseteq Ind(C) \text{ for all } i \in I\}\\ &= Ind(\bigcup\{C \subseteq A \mid Ind(B_i) \subseteq Ind(C) \text{ for all } i \in I\})\\ &= Ind(\bigcap \mathcal{C}_{\mathcal{S}}(B_i)).\end{aligned}$$

(b) The ordered set $(\wp(A)/K_{\mathcal{S}}, \leq)$ has by (4.4) the top element $\top = A/K_{\mathcal{S}}$ and the bottom element $\bot = \emptyset/K_{\mathcal{S}}$. If $\{B_i \mid i \in I\}$ is a subset of $\wp(A)$, then $\bigcap_{i \in I} \mathcal{C}_{\mathcal{S}}(B_i) \in \mathcal{L}_{\mathcal{S}}$. This implies $\mathcal{C}_{\mathcal{S}}(\bigcap_{i \in I} \mathcal{C}_{\mathcal{S}}(B_i)) = \bigcap_{i \in I} \mathcal{C}_{\mathcal{S}}(B_i) \subseteq \mathcal{C}_{\mathcal{S}}(B_i)$ for all $i \in I$. By Proposition 4.7(b) this implies $(\bigcap_{i \in I} \mathcal{C}_{\mathcal{S}}(B_i))/K_{\mathcal{S}} \leq B_i/K_{\mathcal{S}}$ for all $i \in i$. If C/K is a lower bound for $\{B_i/K_{\mathcal{S}} \mid i \in I\}$, then $\mathcal{C}_{\mathcal{S}}(C) \subseteq \mathcal{C}_{\mathcal{S}}(B_i)$ for all $i \in I$. Hence, $\mathcal{C}_{\mathcal{S}}(C) \subseteq \bigcap_{i \in I} \mathcal{C}_{\mathcal{S}}(B_i) = \mathcal{C}_{\mathcal{S}}(\bigcap_{i \in I} \mathcal{C}_{\mathcal{S}}(B_i))$. This implies $C/K \leq (\bigcap_{i \in I} \mathcal{C}_{\mathcal{S}}(B_i))/K_{\mathcal{S}}$. Then $\bigwedge\{B_i/K_{\mathcal{S}} \mid i \in I\} = (\bigcap_{i \in I} \mathcal{C}_{\mathcal{S}}(B_i))/K_{\mathcal{S}}$.

Obviously, $B_i \subseteq \bigcup_{i \in I} B_i$ for all $i \in I$. By (4.4) this implies $B_i/K_{\mathcal{S}} \leq (\bigcup_{i \in I} B_i)/K_{\mathcal{S}}$ for all $i \in I$. If $C/K_{\mathcal{S}}$ is an upper bound for $\{B_i/K_{\mathcal{S}} \mid i \in I\}$, i.e., $B_i/K_{\mathcal{S}} \leq C/K_{\mathcal{S}}$ for all $i \in I$, then by Proposition 4.7(b), $Ind(C) \subseteq Ind(B_i)$ for all $i \in I$. Thus, $Ind(C) \subseteq \bigcap_{i \in I} Ind(B_i) = Ind(\bigcup_{i \in I} B_i)$. By Proposition 4.7(b), this implies $(\bigcup_{i \in I} B_i)/K_{\mathcal{S}} \leq C/K_{\mathcal{S}}$. Hence, $\bigvee\{B_i/K_{\mathcal{S}} \mid i \in I\} = (\bigcup_{i \in I} B_i)/K_{\mathcal{S}}$. □

Now we have shown that the ordered sets $(\{Ind(B) \mid B \subseteq A\}, \supseteq)$, $(\wp(A)/K_{\mathcal{S}}, \leq)$, and $(\mathcal{L}_{\mathcal{S}}, \subseteq)$ are complete lattices. Further, by Proposition 4.7(b) they are order-isomorphic. By Proposition 2.14 we can write the following corollary.

Corollary 4.9 *If $\mathcal{S} = (U, A, \{V_a\}_{a \in A})$ is an information system, then*

$$(\{Ind(B) \mid B \subseteq A\}, \wedge, \vee) \cong (\wp(A)/K_{\mathcal{S}}, \vee, \wedge) \cong (\mathcal{L}_{\mathcal{S}}, \vee, \wedge).$$

The isomorphisms are

$$\begin{aligned} \varphi_1 &: (\wp(A)/K_{\mathcal{S}}, \vee, \wedge) \to (\{Ind(B) \mid B \subseteq A\}, \wedge, \vee), \quad B/K_{\mathcal{S}} \mapsto Ind(B); \\ \varphi_2 &: (\wp(A)/K_{\mathcal{S}}, \vee, \wedge) \to (\mathcal{L}_{\mathcal{S}}, \vee, \wedge); \quad B/K_{\mathcal{S}} \mapsto \mathcal{C}_{\mathcal{S}}(B); \\ \varphi_3 &: (\{Ind(B) \mid B \subseteq A\}, \wedge, \vee) \to (\mathcal{L}_{\mathcal{S}}, \vee, \wedge); \quad Ind(B) \mapsto \mathcal{C}_{\mathcal{S}}(B); \end{aligned}$$

(see Figure 4).

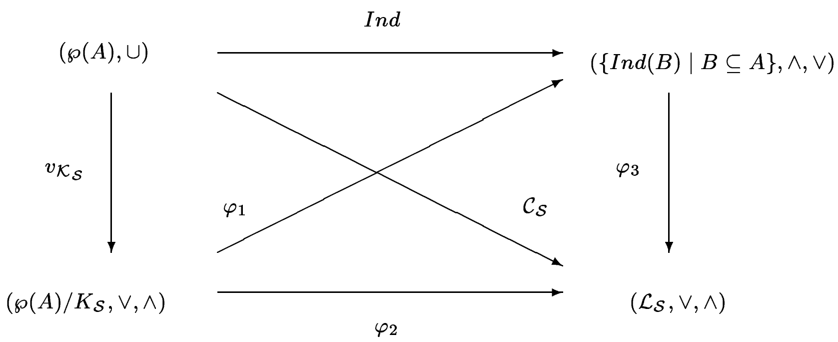

Figure 4

4.3 Independent subsets of attributes and reducts

Here we consider indispensable attributes, independent subsets, cores, and reducts. These notions were studied already in Chapter 3 in the case of knowledge bases. As we have noted before, in an information system, two attributes may define the same classification of objects, and this may cause problems as we see in the following example.

Example 4.10 Let us consider an information system $\mathcal{S} = (U, A, \{V_a\}_{a \in A})$ in which $A = \{a_1, a_2, a_3\}$, and assume $Ind(\{a_1\}) = Ind(\{a_2\})$. In the knowledge base $\mathcal{K}_{\mathcal{S}} = (U, E_{\mathcal{S}})$ the set $E_{\mathcal{S}}$ consists of two equivalences $e_1 = Ind(\{a_1\}) = Ind(\{a_2\})$ and $e_2 = Ind(\{a_3\})$. If $e_1 \not\subseteq e_2$ and $e_2 \not\subseteq e_1$, then both of the equivalences e_1 and e_2 are indispensable in the set $E_{\mathcal{S}}$, but still $Ind(\{a_1, a_2, a_3\}) = Ind(\{a_1, a_3\}) = Ind(\{a_2, a_3\})$, that is, in the set A the attributes a_1, a_2 are dispensable in the sense that the deletion of either of them from A does not change the classification of objects.

From the reason that came out in Example 4.10 we redefine some notions in the case of information systems. Let $\mathcal{S} = (U, A, \{V_a\}_{a \in A})$ be a information system and suppose $B \subseteq A$. An attribute $a \in B$ is *indispensable* in B if $Ind(B) \neq Ind(B - \{a\})$, that is, the classification of objects with respect to B is properly finer than the classification with respect to $B - \{a\}$. The set of all indispensable elements of B is called the *core* of B, and is denoted by $CORE_{\mathcal{S}}(B)$. If $a \in B(\subseteq A)$, then $Ind(B) = Ind(B - \{a\}) \cap Ind(\{a\})$ which implies the useful condition

(4.5) $a(\in B)$ is indispensable in B if and only if $Ind(B - \{a\}) \not\subseteq Ind(\{a\})$.

It is clear that an attribute a is not indispensable in B if the values of the attribute a can be deduced form the set of values of attributes $B - \{a\}$.

The notions of indispensable elements in knowledge bases and in information systems are almost equivalent as we see in the following lemma.

Lemma 4.11 *If $\mathcal{S} = (U, A, \{V_a\}_{a \in A})$ is an information system and $a \in B \subseteq A$, then a is indispensable in B if and only if $Ind(\{a\})$ is indispensable in $\{Ind(\{b\}) \mid b \in B\}$ (in the sense of knowledge bases) and $Ind(\{a\}) \neq Ind(\{b\})$ holds for all $b \in B - \{a\}$.*

Proof. Suppose a is indispensable in B. First, we shall show that $Ind(\{a\}) \neq Ind(\{b\})$ for all $b \in B - \{a\}$. Suppose $Ind(\{a\}) = Ind(\{b\})$ for some $b \in B - \{a\}$. Then $Ind(B - \{a\}) \subseteq Ind(\{b\}) = Ind(\{a\})$, a contradiction! Secondly, assume $Ind(\{a\})$ is not dispensable in $D = \{Ind(\{b\}) \mid b \in B\}$. Let us fix $e \in D$ such that $e = Ind(\{a\})$. Because e is not indispensable in D, we get $Ind(D - \{e\}) \subseteq e$. Because $D - \{e\} \subseteq \{Ind(\{b\}) \mid b \in B - \{a\}\}$, this implies $Ind(B - \{a\}) = \bigcap\{Ind(\{b\}) \mid b \in B - \{a\}\} \subseteq \bigcap(D - \{e\}) = Ind(D - \{e\}) \subseteq e = Ind(\{a\})$, a contradiction!

Conversely, suppose $Ind(\{a\}) \neq Ind(\{b\})$ for all $b \in B - \{a\}$ and $e = Ind(\{a\})$ is indispensable in $D = \{Ind(\{b\}) \mid b \in B\}$. Assume a is not indispensable in B, that is, $Ind(B - \{a\}) \subseteq Ind(\{a\})$. Obviously, $D - \{e\} = \{Ind(\{b\}) \mid b \in B - \{a\}\}$. Then, $Ind(D - \{e\}) = \bigcap(D - \{e\}) = \bigcap\{Ind(\{b\}) \mid b \in B - \{a\}\} = Ind(B - \{a\}) \subseteq Ind(\{a\}) = e$, a contradiction! □

As in the case of knowledge bases, we say that a subset B of attributes A is *independent* if all elements in B are indispensable; otherwise B is *dependent*. The set of all independent subsets of A in $\mathcal{S}$ is denoted by $IND_{\mathcal{S}}$.

By the previous lemma it is clear that a subset of attributes B is independent in an information system $\mathcal{S}$ if and only if the set $\{Ind(\{b\}) \mid b \in B\}$ is independent in the knowledge base $\mathcal{K}_{\mathcal{S}}$ and $Ind(\{a\}) \neq Ind(\{b\})$ for all $a, b \in B$. Also the following condition is obvious.

(4.6) $B \in IND_{\mathcal{S}}$ if and only if $B = CORE_{\mathcal{S}}(B)$.

By Proposition 3.3 and 4.11 we can now write the following lemma.

Lemma 4.12 *If $\mathcal{S} = (U, A, \{V_a\}_{a \in A})$ is an information system and $a \in B \subseteq C(\subseteq A)$, then the following holds.*

(a) *If a is indispensable in C, then a is indispensable in B.*

(b) *If B is independent, then $\{Ind(\{b\}) \mid b \in B\}$ is an antichain, and $Ind(\{a\}) \neq Ind(\{b\})$ for all $a \neq b$ in B.* □

Note that Lemma 4.12(a) follows also from (4.3). It is now obvious that every subset of an independent set is independent, and every superset of a dependent set is dependent. Further, from Lemma 4.12(b) it follows that if $Ind(\{a\}) \subseteq Ind(\{b\})$ for some distinct attributes $a, b \in B$, then B is dependent.

Assume $\mathcal{S} = (U, A, \{V_a\}_{a \in A})$ is an information system and $B \subseteq A$. Then a subset $C \subseteq B$ is said to be a *reduct* of B if C is independent and $Ind(B) = Ind(C)$. The set of all reducts of B in $\mathcal{S}$ is denoted by $RED_{\mathcal{S}}(B)$.

The idea of reducing an attribute set in an information system is of great practical importance, because it shows that one can get sometimes the same information from the system with a smaller set of attributes. Now it is obvious that C is a reduct of B in an information system $\mathcal{S}$ if and only if $\{Ind(\{c\}) \mid c \in C\}$ is a reduct of $\{Ind(\{b\}) \mid b \in B\}$ in the knowledge base $\mathcal{K}_{\mathcal{S}}$ and $Ind(\{a\}) \neq Ind(\{b\})$ for all distinct $a, b \in C$.

In section 3.2 we gave a condition under which every subset of a knowledge base has a finite reduct. For a knowledge base $\mathcal{K}$ and an information system $\mathcal{S}$ we define the following sets.

$$\begin{aligned} \mathcal{F}_{\mathcal{K}} &= \{Ind_{\mathcal{K}}(F) \mid F \text{ is a finite subset of } E\}, \text{ and} \\ \mathcal{F}_{\mathcal{S}} &= \{Ind_{\mathcal{S}}(F) \mid F \text{ is a finite subset of } A\}, \end{aligned}$$

where $Ind_{\mathcal{K}}(F)$ (resp. $Ind_{\mathcal{S}}(F)$) refers that the indiscernibility relation in defined in the knowledge base $\mathcal{K}$ (resp. information system $\mathcal{S}$). The next lemma is trivial.

Lemma 4.13 *If $\mathcal{S} = (U, A, \{V_a\}_{a \in A})$ is an information system and $\mathcal{K}_{\mathcal{S}} = (U, E_{\mathcal{S}})$, where $E_{\mathcal{S}} = \{Ind(\{a\}) \mid a \in A\}$, then*

$$\mathcal{F}_{\mathcal{K}_{\mathcal{S}}} = \mathcal{F}_{\mathcal{S}}.$$

□

By Lemma 4.13 it is clear that the set $\mathcal{F}_{\mathcal{S}}$ satisfies the DCC if and only if $\mathcal{F}_{\mathcal{K}_{\mathcal{S}}}$ satisfies the DCC. Therefore, if $\mathcal{F}_{\mathcal{S}}$ satisfies the DCC, then by Lemma 3.4, for any $B \subseteq A$, there exists a finite $C \subseteq B$ which satisfies $Ind(B) = Ind(C)$. Moreover, the following proposition can be easily verified by Proposition 3.5.

Proposition 4.14 *Suppose $\mathcal{S} = (U, A, \{V_a\}_{a \in a})$ is an information system and let $\mathcal{F} = \{Ind(F) \mid F$ is a finite subset of $A\}$.*

(a) *If $\mathcal{F}$ satisfies the* DCC, *then every subset of A has a finite reduct.*

(b) *If $\mathcal{F}$ satisfies the* DCC, *then $CORE_{\mathcal{S}}(B) = \bigcap RED_{\mathcal{S}}(B)$ for all $B \subseteq A$.*

(c) *If $\{Ind(\{a\}) \mid a \in A\}$ has no minimal element with respect to the inclusion relation, then there exists $B(\subseteq A)$ which has no reducts; moreover $CORE_{\mathcal{S}}(B) \neq \bigcap RED_{\mathcal{S}}(B)$ holds.* □

Next we present a lemma which guarantees that $\mathcal{F}_{\mathcal{S}}$ satisfies the DCC.

Lemma 4.15 *Suppose $\mathcal{S} = (U, A, \{V_a\}_{a \in A})$ is an information system. If at least one of the sets U or A is finite, then $\mathcal{F}_{\mathcal{S}}$ satisfies the* DCC.

Proof. It is obvious that if U is finite, then $\mathrm{Eq}(U)$ is finite. Because $\mathcal{F}_{\mathcal{S}} \subseteq \mathrm{Eq}(U)$, this implies that $\mathcal{F}_{\mathcal{S}}$ if finite, and hence it satisfies the DCC. Secondly, if A is finite, then $\wp(A)$ is finite. Because the mapping $Ind : \wp(A) \to \{Ind(B) \mid B \subseteq A\}$ is onto, obviously $|\mathcal{F}_{\mathcal{S}}| \leq |\wp(A)|$ holds, which implies that $\mathcal{F}_{\mathcal{S}}$ is finite. □

Note that if we require that for all $a \in A$, the set V_a — or even that the set $V = \bigcup_{a \in A} V_a$ is finite, this does not imply that $\mathcal{F}_{\mathcal{S}}$ satisfies the DCC. We shall give an example which illustrates this.

Example 4.16 Suppose $\mathcal{S} = (U, A, \{V_a\}_{a \in A})$ is an information system such that $U = \{x_i \mid i \in \mathbb{N}\}$, $A = \{a_i \mid i \in \mathbb{N}\}$, and $V_a = \{0, 1\}$ for all $a \in A$. It is clear that also the set $V = \bigcup_{a \in A} V_a = \{0, 1\}$ is finite. For each $i \in \mathbb{N}$, the attribute a_i is defined by

$$a_i(x_j) = \begin{cases} 0 & \text{if } j \leq i, \\ 1 & \text{otherwise.} \end{cases}$$

The equivalence classes of $Ind(\{a_i\})$, $i \in \mathbb{N}$, are $\{x_1, \ldots, x_i\}$ and $\{x_{i+1}, x_{i+2}, \ldots\}$. Obviously,

$$Ind(\{a_1\}) \supset Ind(\{a_1, a_2\}) \supset \cdots \supset Ind(\{a_1, \ldots, a_k\}) \supset Ind(\{a_1, \ldots, a_{k+1}\}) \supset \cdots$$

is an infinite descending chain in $\mathcal{F}_{\mathcal{S}}$.

By Pawlak's original definition, in an information system all the sets U, A, and $V = \bigcup_{a \in A} V_a$ are assumed to be finite. Proposition 4.12 says that if either of the set U or A is finite, then every subset B of A has a finite reduct. By example 4.16, the finiteness of the sets V_a or even the set V does not guarantee this.

If $\mathcal{F}_{\mathcal{S}}$ satisfies the DCC, then we can form a finite reduct of a subset B of attributes by determining a finite reduct D of $\{Ind(\{b\}) \mid b \in B\}$ in the knowledge base $\mathcal{K}_{\mathcal{S}}$ by applying the method presented in the proof of Proposition 3.5(a), and then for all $e \in D$, we may choose exactly one $a \in B$ which satisfies $Ind(\{a\}) = e$.

In Chapter 5 we shall study discernibility matrices. We shall see how for any information system such that the sets U and V are finite we can, for example, compute the set of *all* reducts of a subset of attributes.

4.4 Dependency relations in information systems

Dependencies in an information system are basic tools for drawing conclusions from the basic knowledge. Namely, often the value of some attribute for an object can derived from the values of some other attributes. For example, if the value of an attribute "age" is "two years", then the value of an attribute "education" will be "no education".

Let $\mathcal{S} = (U, A, \{V_a\}_{a \in A})$ be an information system. We say that a set $B(\subseteq A)$ of attributes *depends on* a set $C(\subseteq A)$ of attributes in $\mathcal{S}$, denoted by $C \to B\ (\mathcal{S})$, if $Ind(C) \subseteq Ind(B)$. The relation $\to\ (\mathcal{D})$ is called the *dependency relation of* $\mathcal{S}$. Usually, we write simply $C \to B$ if there is no danger of confusion.

It is clear that if $C \to B$, then any two objects discernible by B are also discernible by C. Hence, we can use the relation $\to$ for deducing. If objects x and y are indiscernible with respect to C and $C \to B$, then x and y are B-indiscernible.

By (4.3), $B \subseteq C$ implies $C \to B$. Such dependencies are called *trivial* (see e.g. [25]). It is clear that $Ind(B) = Ind(C)$ if and only if $B \to C$ and $C \to B$. The following obvious lemma says that the notions of dependency in $\mathcal{S}$ and in the corresponding knowledge base $\mathcal{K}_{\mathcal{S}}$ are essentially equivalent.

Lemma 4.17 *Let $\mathcal{S} = (U, A, \{V_a\}_{a \in A})$ be an information system and $B, C \subseteq A$. Then the following conditions are equivalent.*

(a) $C \to B$ $(\mathcal{S})$.

(b) $\{Ind(\{c\}) \mid c \in C\} \to \{Ind(\{b\}) \mid b \in B\}$ $(\mathcal{K}_{\mathcal{S}})$. □

Dependencies have an important role in the special class of information systems called *decision tables*. Decision tables [21] are information systems with the set of attributes divided into two disjoint sets Con and Dec, called *condition* and *decision attributes*, respectively. A decision table $\mathcal{S}$ is called *consistent* if $Con \to Dec$ $(\mathcal{S})$ holds, that is, the values of the decision attributes are really determined by the condition attributes.

Example 4.18 Let us consider the information system of Example 4.1 and assume that $Con = \{1, 2, 3\}$ and $Dec = \{4\}$. It is clear that $Con \to Dec$, that is, the decision table $\mathcal{S}$ is consistent. It can be easily verified that also $\{1, 2\} \to \{4\}$ holds. Hence, the attributes 1 and 2 totally determine the value of the decision attribute 4. Note that the set $\{1, 2\}$ is not a reduct of the set $\{1, 2, 3\}$.

Especially from the point of view of decision tables, the following problem is important. Suppose B and C are subsets of attributes of an information system such that $C \to B$ holds. Then find all minimal subsets D of C which satisfy $D \to B$. In Section 5.3 we shall present a solution to this problem by applying discernibility matrices which is different from the solution appearing in [26].

Chapter 5

Discernibility matrices and functions

5.1 Discernibility matrices

The notion of discernibility matrices is introduced in [26]. By applying discernibility matrices we may write several algorithms for computing e.g. the reducts of a subset of attributes in an information system. We note that the results of Lemma 5.1 and Propositions 5.6(b) and 5.7 can be found also from [26] in a slightly different form.

Suppose $\mathcal{S} = (U, A, \{V_a\}_{a \in A})$ is an information system in which $U = \{x_1, \ldots, x_n\}$. Let us define an $n \times n$-matrix (c_{ij}), called the *discernibility matrix* of $\mathcal{S}$, by

$$c_{ij} = \{a \in A \mid a(x_i) \neq a(x_j)\}$$

for all $1 \leq i, j \leq n$.

Obviously, $c_{ij} = c_{ji}$ for all $1 \leq i, j \leq n$ and $c_{ii} = \emptyset$ for all $1 \leq i \leq n$. Therefore, we can represent $(c_{ij})_{n \times n}$ by the elements in the upper triangle of (c_{ij}) only, i.e., by the elements c_{ij} with $1 \leq i < j \leq n$.

Now it is quite easy to decide for any subset $B (\subseteq A)$ of attributes whether two objects x_i and x_j are B-indiscernible.

Lemma 5.1 *If $\mathcal{S} = (U, A, \{V_a\}_{a \in A})$ is an information system such that $U = \{x_1, \ldots, x_n\}$, $(c_{ij})_{n \times n}$ is the discernibility matrix of $\mathcal{S}$, and $B \subseteq A$, then for all $1 \leq i < j \leq n$, $(x_i, x_j) \in Ind(B)$ if and only if $c_{ij} \cap B = \emptyset$.*

Proof. If $(x_i, x_j) \in Ind(B)$, then $a(x_i) = a(x_j)$ for all $a \in B$. This implies $a \notin c_{ij}$ for all $a \in B$, i.e., $c_{ij} \cap B = \emptyset$. Conversely, if $c_{ij} \cap B = \emptyset$, then $a \notin c_{ij}$ holds for all $a \in B$. Hence for all $a \in B$, $a(x_i) = a(x_j)$, that is, $(x_i, x_j) \in Ind(B)$. □

By our following lemma we can test if the classification of objects with respect to a subset of attributes is finer than or equal to the classification of objects with respect to another subset of attributes.

Lemma 5.2 *If $\mathcal{S} = (U, A, \{V_a\}_{a \in A})$ is an information system such that $U = \{x_1, \ldots, x_n\}$, $(c_{ij})_{n \times n}$ is the discernibility matrix of $\mathcal{S}$, and $B, C \subseteq A$, then the following conditions are equivalent.*

(a) $Ind(C) \subseteq Ind(B)$.
(b) *For all* $1 \leq i < j \leq n$, $c_{ij} \cap B \neq \emptyset$ *implies* $c_{ij} \cap C \neq \emptyset$.

Proof. Assume $Ind(C) \subseteq Ind(B)$. If $c_{ij} \cap B \neq \emptyset$ for some $1 \leq i < j \leq n$, then by Lemma 5.1, $(x_i, x_j) \notin Ind(B)$. This implies $(x_i, x_j) \notin Ind(C)$ and $c_{ij} \cap C \neq \emptyset$. On the other hand, assume $c_{ij} \cap B \neq \emptyset$ implies $c_{ij} \cap C \neq \emptyset$ for all $1 \leq i < j \leq n$. Suppose $(x_i, x_j) \in Ind(C)$ for some $1 \leq i, j \leq n$. If $i = j$, then trivially $(x_i, x_j) \in Ind(B)$. If $i < j$, then $c_{ij} \cap C = \emptyset$, which implies $c_{ij} \cap B = \emptyset$, i.e., $(x_i, x_j) \in Ind(B)$. If $i > j$, then $C \cap c_{ji} = \emptyset$ from which we get $B \cap c_{ji} = \emptyset$. Therefore, $(x_j, x_i) \in Ind(B)$ and $(x_i, x_j) \in Ind(B)$, because $Ind(B)$ is symmetric. □

Next we present two simple corollaries of the previous lemma. The first is based on the trivial fact that $Ind(B) = Ind(C)$ if and only if $Ind(B) \subseteq Ind(C)$ and $Ind(C) \subseteq Ind(B)$, and the second on (4.3) by which $C \subseteq B$ implies $Ind(B) \subseteq Ind(C)$.

Corollary 5.3 *Suppose* $\mathcal{S} = (U, A, \{V_a\}_{a \in A})$ *is an information system such that* $U = \{x_1, \ldots, x_n\}$. *If* $(c_{ij})_{n \times n}$ *is the discernibility matrix of* $\mathcal{S}$ *and* $B, C \subseteq A$, *then the following conditions are equivalent.*
(a) $Ind(B) = Ind(C)$.
(b) *For all* $1 \leq i < j \leq n$, $c_{ij} \cap B = \emptyset$ *if and only if* $c_{ij} \cap C = \emptyset$.

Corollary 5.4 *Suppose* $\mathcal{S} = (U, A, \{V_a\}_{a \in A})$ *is an information system such that* $U = \{x_1, \ldots, x_n\}$. *If* $(c_{ij})_{n \times n}$ *is the discernibility matrix of* $\mathcal{S}$ *and* $C \subseteq B (\subseteq A)$, *then the following conditions are equivalent.*
(a) $Ind(B) = Ind(C)$.
(b) *For all* $1 \leq i < j \leq n$, $c_{ij} \cap B \neq \emptyset$ *implies* $c_{ij} \cap C \neq \emptyset$.

Example 5.5 Let us consider the information system $\mathcal{S}$ of Example 4.1. Its discernibility matrix (c_{ij}), $1 \leq i < j \leq 4$, is presented in Table 2.

	2	3	4
1	$\{2,4\}$	$\{1,2,4\}$	$\{2,3,4\}$
2		$\{1,4\}$	$\{3\}$
3			$\{1,3,4\}$

Table 2

By applying discernibility matrices it is easy to decide whether an attribute is indispensable in a subset of attributes or not. Our following proposition characterizes the set $IND_{\mathcal{S}}$, and for all $B \subseteq A$ the set $CORE_{\mathcal{S}}(B)$.

Proposition 5.6 *Suppose* $\mathcal{S} = (U, A, \{V_a\}_{a \in A})$ *is an information system such that* $U = \{x_1, \ldots, x_n\}$. *If* $(c_{ij})_{n \times n}$ *is the discernibility matrix of* $\mathcal{S}$ *and* $B \subseteq A$, *then the following equations hold.*
(a) $IND_{\mathcal{S}} = \{B \subseteq A \mid$ *for all* $a \in B,\ c_{ij} \cap B = \{a\}$ *for some* $1 \leq i < j \leq n\}$.
(b) $CORE_{\mathcal{S}}(B) = \{a \in B \mid c_{ij} \cap B = \{a\}$ *for some* $1 \leq i < j \leq n\}$.

Proof. Assume $B \subseteq A$ and $a \in B$. By Corollary 5.4, $B - \{a\} \subseteq B$ implies that $Ind(B) \neq Ind(B - \{a\}) \Leftrightarrow$ there exist $1 \leq i < j \leq n$, such that $c_{ij} \cap B \neq \emptyset$ and $(B - \{a\}) \cap c_{ij} = \emptyset \Leftrightarrow c_{ij} \cap B = \{a\}$ for some $1 \leq i < j \leq n$. This equivalence implies statements (a) and (b). □

The following proposition characterizes the reducts of a given subset of attributes.

Proposition 5.7 *Suppose $\mathcal{S} = (U, A, \{V_a\}_{a \in A})$ is an information system such that $U = \{x_1, \ldots, x_n\}$ and $(c_{ij})_{n \times n}$ is the discernibility matrix of $\mathcal{S}$. If $B \subseteq A$, then $C \in RED_{\mathcal{S}}(B)$ if and only if C is minimal with respect to inclusion among the subsets of A such that $C \cap (c_{ij} \cap B) \neq \emptyset$ for all $1 \leq i < j \leq n$ which satisfy $c_{ij} \cap B \neq \emptyset$.*

Proof. Suppose $C \in RED_{\mathcal{S}}$. Then $C \subseteq B$ and $Ind(B) = Ind(C)$. By Corollary 5.4, $C \cap (c_{ij} \cap B) = c_{ij} \cap (B \cap C) = c_{ij} \cap C \neq \emptyset$ whenever $c_{ij} \cap B \neq \emptyset$. If C is not minimal, there is a $C_1 \subset C$ such that $C_1 \cap (c_{ij} \cap B) \neq \emptyset$ whenever $c_{ij} \cap B \neq \emptyset$. But $C_1 \subset C \subseteq B$ implies $C_1 \cap (c_{ij} \cap B) = c_{ij} \cap (B \cap C_1) = c_{ij} \cap C_1$. So, $Ind(C_1) = Ind(B) = Ind(C)$, a contradiction!

Conversely, let C be a minimal subset of A which satisfies $C \cap (c_{ij} \cap B) \neq \emptyset$ whenever $c_{ij} \cap B \neq \emptyset$. If $C \not\subseteq B$, then $C_1 = (B \cap C) \subset C$ and $C_1 \cap (c_{ij} \cap B) = (B \cap C) \cap (c_{ij} \cap B) = C \cap (c_{ij} \cap B) \neq \emptyset$ whenever $c_{ij} \cap B \neq \emptyset$, a contradiction! So, $C \subseteq B$. Since $c_{ij} \cap C = C \cap (c_{ij} \cap B)$, we get $Ind(B) = Ind(C)$. Suppose that C is not independent. Then there is a $C_1 \subset C$ such that $Ind(C_1) = Ind(C) = Ind(B)$. Because $C_1 \subseteq B$, this implies $C_1 \cap (c_{ij} \cap B) = c_{ij} \cap C_1 \neq \emptyset$ whenever $c_{ij} \cap B \neq \emptyset$, a contradiction! □

Note that Proposition 5.7 characterizes the reducts of B as subsets of A (instead of B which would be a more natural way). We shall need this particular characterization later when we are writing an algorithm which computes $RED_{\mathcal{S}}(B)$ for an arbitrary set $B(\subseteq A)$ of attributes.

5.2 Discernibility functions

In this section we study the notion of discernibility functions which helps us to write algorithms for e.g. the reduction problem. Note that in [26] the discernibility function is defined only for the set A in the information system $\mathcal{S} = (U, A, \{V_a\}_{a \in A})$, but here we define it for any arbitrary subset B of A.

First we shall recall some notions concerning Boolean logic (see e.g. [1, 9, 18]). Let us fix a countable infinite alphabet of *Boolean variables* $\{x_1, x_2, \ldots\}$. We can combine Boolean variables using *Boolean connectives* such as $\vee$ (*logical or*), $\wedge$ (*logical or*) and $\neg$ (*logical not*). A *Boolean expression* can be any of (a) a Boolean variable, (b) an expression of the form $\neg\phi_1$, where ϕ_1 is a Boolean expression, (c) an expression of the form $(\phi_1 \vee \phi_2)$, where ϕ_1 and ϕ_2 are Boolean expressions, or (d) an expression of the form $(\phi_1 \wedge \phi_2)$, where ϕ_1 and ϕ_2 are Boolean expressions. In case (b) the expression is called the *negation* of ϕ_1; in case (c) it is the *disjunction* of ϕ_1 and ϕ_2; and in case (d) it is the *conjunction* of ϕ_1 and ϕ_2. An expression of the form x_i or $\neg x_i$ is called a *literal*. A conjunction of literals is called a $\wedge$-*term*.

An m-*ary Boolean function*, or a *function* for short, is a mapping $f : \{0,1\}^m \to \{0,1\}$. An element $v \in \{0,1\}^m$ is called a *Boolean vector* (a *vector* for short). It is known that each Boolean expression expresses some Boolean function and any m-ary Boolean function f can be expressed as a Boolean expression ϕ_f involving variables $x_1, \ldots, x_m$ (see [18], for example).

If $f(v) = 1$ (resp. 0), then v is called a *true* (resp. *false*) vector of f. The set of all true vectors (false vectors) is denoted by $T(f)$ $(F(f))$. We denote by $\perp$ and $\top$ the two special functions for which $T(\perp) = \emptyset$ and $F(\top) = \emptyset$, respectively. Moreover for all $m \in \mathbb{N}$, we write

$$\mathbf{0}_m = (\underbrace{0, \ldots, 0}_{m}) \qquad \text{and} \qquad \mathbf{1}_m = (\underbrace{1, \ldots, 1}_{m}).$$

Let $u = (u_1, \ldots, u_m)$ and $v = (v_1, \ldots, v_m)$ be vectors. We set $u \leq v$ if and only if $u_i \leq v_i$, for $1 \leq i \leq m$. A function f is *isotone* if $v \leq w$ always implies $f(v) \leq f(w)$. In the sequel we shall assume that f is a isotone function. A true vector v of f is *minimal* if there is no true vector w such that $w < v$, and let $\min T(f)$ denote the set of all minimal true vectors of f. A *maximal* false vector is defined dually and $\max F(f)$ denotes the set of all maximal false vectors of f. The following facts are obvious.

$$T(f) = \{v \mid v \geq w \text{ for some } w \in \min T(f)\}.$$
$$F(f) = \{v \mid v \leq w \text{ for some } w \in \max F(f)\}.$$

Two vectors u and v are *incomparable* if neither $u \leq v$ nor $u \geq v$ holds. A set of vectors $V(\subseteq \{0,1\}^m)$ is called *incomparable* if every pair of distinct vectors $u, v \in V$ is incomparable. Obviously, the sets $\min T(f)$ and $\max F(f)$ are both incomparable.

Let f and g be Boolean functions. If $g(v) \leq f(v)$ for all $v \in \{0,1\}^m$, then we say that g *implies* f. An *implicant* of a Boolean function f is a $\wedge$-term which implies f. A $\wedge$-term ϕ_1 is said to *subsume* ϕ_2 if all literals of ϕ_2 are literals of ϕ_1. A *prime implicant* of f is defined as an implicant of f such that no $\wedge$-term subsumed by it by it can be an implicant of f. An *irredundant disjunctive normal* form of f is a disjunction of prime implicants of f such that a removal of any of them makes the remaining expression no longer equivalent to the original f. It is known that if f is isotone, it has a unique irredundant disjunctive normal form consisting of all prime implicants of f. Moreover, there is a bijective correspondence between the prime implicants and the minimal true vectors of a isotone function (see [9], for example).

Suppose that $\mathcal{S} = (U, A, \{V_a\}_{a\in A})$ is an information system in which $U = \{x_1, \ldots, x_n\}$ and $A = \{a_1, \ldots, a_m\}$. For any $B \subseteq A$, let $\delta(B)$ denote the disjunction of all variables y_k, where $a_k \in B$. We define the *discernibility function* $f_B^{\mathcal{S}}(y_1, \ldots, y_m)$ of a subset $B(\subseteq A)$ as the conjunction

$$\bigwedge_{\substack{1\leq i<j\leq n,\\ c_{ij}\cap B\neq\emptyset}} \delta(c_{ij} \cap B).$$

Obviously, the function $f_B^{\mathcal{S}}$ is isotone. Because the empty conjunction $\bigwedge \emptyset = 1$, $f_B^{\mathcal{S}} = \top$ if and only if $c_{ij} \cap B = \emptyset$ for all $1 \leq i < j \leq n$ if and only if $Ind(B) = U^2$. A function $\chi : \wp(A) \to \{0,1\}^m$ is defined by

$$B \mapsto (\chi_1(B), \ldots, \chi_m(B)),$$

where

$$\chi_k(B) = \begin{cases} 0 & \text{if } a_k \notin B \\ 1 & \text{if } a_k \in B \end{cases}$$

for all $k,\ 1 \leq k \leq m$. The value $\chi(B)$ is called the *characteristic vector* of B.

Let us denote $B' = A - B$ for any $B \subseteq A$. Because for all $B, C \subseteq A$, $B \cap C = \emptyset$ if and only if $B \subseteq C'$, by the definition of the function $f_B^{\mathcal{S}}$ we can now write the following conditions for every $B, C(\subseteq A)$.

(5.1) $f_B^{\mathcal{S}}(\chi(C)) = 1 \Leftrightarrow C \cap (c_{ij} \cap B) \neq \emptyset$ for all $1 \leq i < j \leq n$ which satisfy $c_{ij} \cap B \neq \emptyset$.

(5.2) $f_B^{\mathcal{S}}(\chi(C)) = 0 \Leftrightarrow C \subseteq (c_{ij} \cap B)'$ for some $1 \leq i < j \leq n$ which satisfy $c_{ij} \cap B \neq \emptyset$.

Our following proposition follows easily from (5.1), (5.2), and Lemma 5.7.

Proposition 5.8 *Suppose $\mathcal{S} = (U, A, \{V_a\}_{a\in A})$ is an information system such that $U = \{x_1, \ldots, x_n\}$, $A = \{a_1, \ldots, a_m\}$, and $(c_{ij})_{n\times n}$ is the discernibility matrix of $\mathcal{S}$.*

(a) $\min T(f_B^{\mathcal{S}}) = \{\chi(C) \mid C \in RED_{\mathcal{S}}(B)\}$.

(b) $\max F(f_B^{\mathcal{S}}) = \max\{\chi((c_{ij} \cap B)') \mid 1 \leq i < j \leq n,\ c_{ij} \cap B \neq \emptyset\}$. □

Note that by Proposition 5.8 we can compute the set of all reducts of B by identifying the set of minimal true vectors of the function $f_B^{\mathcal{S}}$. This observation is used in Section 5.4 where we shall present a selection of algorithms.

Corollary 5.9 *Suppose $\mathcal{S} = (U, A, \{V_a\}_{a\in A})$ is an information system such that $U = \{x_1, \ldots, x_n\}$, $A = \{a_1, \ldots, a_m\}$, and $(c_{ij})_{n\times n}$ is the discernibility matrix of $\mathcal{S}$. If $B \subseteq A$, then the set $\{a_{i_1}, \ldots, a_{i_p}\}$ is a reduct of B if and only if $y_{i_1} \wedge \cdots \wedge y_{i_p}$ is a prime implicant of $f_B^{\mathcal{S}}$.*

Example 5.10 Let us consider the information system $\mathcal{S}$ of Example 4.1. Its discernibility matrix (c_{ij}) is presented in Example 5.5. The discernibility function of the set A is

$$\begin{aligned} f_A^{\mathcal{S}} &= (2 \vee 4) \wedge (1 \vee 2 \vee 4) \wedge (2 \vee 3 \vee 4) \wedge (1 \vee 4) \wedge 3 \wedge (1 \vee 3 \vee 4) \\ &= 3 \wedge (1 \vee 4) \wedge (2 \vee 4) = 3 \wedge (4 \vee (1 \wedge 2)) \\ &= (3 \wedge 4) \vee (1 \wedge 2 \wedge 3), \end{aligned}$$

where i stands for y_i. The function $f_A^{\mathcal{S}}$ has obviously the prime implicants $(3 \wedge 4)$ and $(1 \wedge 2 \wedge 3)$, which implies that $RED_{\mathcal{S}}(A) = \{\{3, 4\}, \{1, 2, 3\}\}$.

5.3 Dependency relations and dependency functions

Suppose $\mathcal{S} = (U, A, \{V_a\}_{a\in A})$ is an information system such that $U = \{x_1, \ldots, x_n\}$. If $(c_{ij})_{n\times n}$ is the discernibility matrix of $\mathcal{S}$, then by Lemma 5.2 the following condition holds for all $B, C \subseteq A$.

(5.3) $\quad C \to B$ if and only if $c_{ij} \cap B \neq \emptyset$ implies $c_{ij} \cap C \neq \emptyset$ for all $1 \leq i < j \leq n$.

In Section 4.4 we presented the following problem. Let B and C be subsets of A which satisfy $C \to B$. Find the minimal subsets D of C such that $D \to B$. In [26] this problem is solved by applying the notions of discernibility functions and lower approximations of subsets of objects which are studied in the theory of rough sets (see e.g. [20, 27, 21]). Here we present a more natural solution. Our following proposition characterizes the sets mentioned above.

Proposition 5.11 *Suppose $\mathcal{S} = (U, A, \{V_a\}_{a\in A})$ is an information system such that $U = \{x_1, \ldots, x_n\}$ and $(c_{ij})_{n\times n}$ is the discernibility matrix of $\mathcal{S}$. If $C \to B$ $(\mathcal{S})$ holds, then the following conditions are equivalent.*

(a) *D is a minimal subset of C such that $D \to B$.*

(b) *D is a minimal subset of A such that $c_{ij} \cap B \neq \emptyset$ implies $D \cap (c_{ij} \cap C) \neq \emptyset$ for all $1 \leq i < j \leq n$.*

Proof. Suppose $C \to B$ and assume D is a minimal subset of C such that $D \to B$. Because $D \subseteq C$, we get $D \cap (c_{ij} \cap C) = c_{ij} \cap (C \cap D) = c_{ij} \cap D$. Now the assumption $D \to B$ implies that $D \cap (c_{ij} \cap C) \neq \emptyset$ holds for all $1 \leq i < j \leq n$ which satisfy $c_{ij} \cap B \neq \emptyset$. If D is not minimal, there is a $D_1 \subset D$ such that $D_1 \cap (c_{ij} \cap C) \neq \emptyset$ whenever $c_{ij} \cap B \neq \emptyset$. But $D_1 \subseteq C$ implies $D_1 \cap (c_{ij} \cap C) = c_{ij} \cap D_1$. So, $D_1 \to B$, a contradiction!

Conversely, suppose $C \to B$ and D is a minimal subset of A which satisfies $D \cap (c_{ij} \cap C) \neq \emptyset$ whenever $c_{ij} \cap B \neq \emptyset$. If $D \not\subseteq C$, then $D_1 = (C \cap D) \subset C$, and $D_1 \cap (c_{ij} \cap C) = (C \cap D) \cap (c_{ij} \cap C) = D \cap (c_{ij} \cap C) \neq \emptyset$ whenever $c_{ij} \cap B \neq \emptyset$, a contradiction! So, $D \subseteq C$. Since $D \cap (c_{ij} \cap C) = c_{ij} \cap D$, we get $D \to B$. Assume there a $D_1 \subset D$ such that $D_1 \to B$, i.e., $c_{ij} \cap D_1 \neq \emptyset$ whenever $c_{ij} \cap B \neq \emptyset$. Because $D_1 \subset D \subseteq C$, this implies $D_1 \cap (c_{ij} \cap C) = c_{ij} \cap D_1 \neq \emptyset$ whenever $c_{ij} \cap B \neq \emptyset$, a contradiction! $\square$

Let $\mathcal{S} = (U, A, \{V_a\}_{a\in A})$ be an information system in which $U = \{x_1, \ldots, x_n\}$ and $A = \{a_1, \ldots, a_m\}$. If $(c_{ij})_{n\times n}$ is the discernibility matrix of $\mathcal{S}$ and B, C are subsets of A which satisfy $C \to B$ in $\mathcal{S}$, then we define the *dependency function* $f^{\mathcal{S}}_{C\to B}(y_1, \ldots, y_m)$ of the dependency $C \to B$ as the conjunction

$$\bigwedge_{\substack{1\le i<j\le n,\\ c_{ij}\cap B\neq\emptyset}} \delta(c_{ij} \cap C).$$

Obviously, the function $f^{\mathcal{S}}_{C\to B}$ is isotone and $f^{\mathcal{S}}_{C\to B} = \top$ if and only if $Ind(B) = U^2$. By the definition of $f^{\mathcal{S}}_{C\to B}$ we can now write the following conditions for all subsets B, C and D of A which satisfy $C \to B$,

(5.4) $f^{\mathcal{S}}_{C\to B}(\chi(D)) = 1 \Leftrightarrow D \cap (c_{ij} \cap C) \neq \emptyset$ for all $1 \le i < j \le n$ such that $c_{ij} \cap B \neq \emptyset$.

(5.5) $f^{\mathcal{S}}_{C\to B}(\chi(D)) = 0 \Leftrightarrow D \subseteq (c_{ij} \cap C)'$ for some $1 \le i < j \le n$ such that $c_{ij} \cap B \neq \emptyset$.

By (5.4), (5.5), and Proposition 5.11 we can now present the following proposition.

Proposition 5.12 *Suppose $\mathcal{S} = (U, A, \{V_a\}_{a\in A})$ is an information system such that $U = \{x_1, \ldots, x_n\}$, $A = \{a_1, \ldots, a_m\}$, and $(c_{ij})_{n\times n}$ is the discernibility matrix of $\mathcal{S}$. If the dependency $C \to B$ $(\mathcal{S})$ holds, then the following equations are valid.*

(a) $\min T(f^{\mathcal{S}}_{C\to B}) = \{\chi(D) \mid D$ *is a minimal subset of* C *such that* $D \to B\}$.

(b) $\max F(f^{\mathcal{S}}_{C\to B}) = \max\{\chi((c_{ij} \cap C)') \mid 1 \le i < j \le n,\ c_{ij} \cap B \neq \emptyset\}$.

By Proposition 5.12 we can compute for the dependency $C \to B$ the set of all minimal subsets D of C such that $D \to B$ by identifying the set of all minimal true vectors of $f^{\mathcal{S}}_{C\to B}$.

Corollary 5.13 *Assume $\mathcal{S} = (U, A, \{V_a\}_{a\in A})$ is an information system such that $U = \{x_1, \ldots, x_n\}$, $A = \{a_1, \ldots, a_m\}$, and $(c_{ij})_{n\times n}$ is the discernibility matrix of $\mathcal{S}$. If $C \to B$ holds in $\mathcal{S}$, then $D = \{a_{i_1}, \ldots, a_{i_p}\}$ is a minimal subset of C which satisfies $D \to B$ if and only if $y_{i_1} \wedge \cdots \wedge y_{i_p}$ is a prime implicant of $f^{\mathcal{S}}_{C\to B}$.*

Example 5.14 Let us consider the information system $\mathcal{S}$ of Example 4.1. Its discernibility matrix (c_{ij}) is presented in Example 5.5. If we set $B = \{4\}$, then the trivial dependency $A \to B$ holds in $\mathcal{S}$. The dependency function of the dependency $A \to B$ is

$$\begin{aligned} f^{\mathcal{S}}_{A\to B} &= (2 \vee 4) \wedge (1 \vee 2 \vee 4) \wedge (2 \vee 3 \vee 4) \wedge (1 \vee 4) \wedge (1 \vee 3 \vee 4) \\ &= (1 \vee 4) \wedge (2 \vee 4) \\ &= 4 \vee (1 \wedge 2), \end{aligned}$$

where i stands for y_i. The function $f^{\mathcal{S}}_{A\to B}$ has obviously the prime implicants 4 and $(1 \wedge 2)$, which implies that $\{4\}$ and $\{1, 2\}$ are the minimal subsets D of A which satisfy $D \to B$.

5.4 A data type and basic algorithms for discernibility matrices

In this section we present a simple implementation of discernibility matrices as a data type, which is sufficient for us to solve problems concerning, cores, dependencies, independent sets, and reducts in an information system.

The discernibility matrix could be given as an $n \times n$–matrix in which the entry (i, j) is the vector $\chi(c_{ij})$, but as we have seen, only the entries c_{ij}, where $1 \leq i < j \leq n$, are needed. Therefore, we introduce the following representation which saves always over half of the space compared to the matrix representation.

Let $\mathcal{S} = (U, A, \{V_a\}_{a \in A})$ be an information system such that $U = \{x_1, \ldots, x_n\}$ and $A = \{a_1, \ldots, a_m\}$. Then the discernibility matrix $(c_{ij})_{n \times n}$ of $\mathcal{S}$ can be represented as an array $c[1..n(n-1)/2]$ of length $n(n-1)/2$, in which

$$c[k] = \chi(c_{ij}),$$

where $k = j(j-1)/2 - i + 1$ for all $1 \leq i < j \leq n$.

Example 5.15 The discernibility matrix of Example 5.5 can be represented as an array c[1..6] in which

$$\begin{gathered} c[1] = \chi(c_{12}) = (0,1,0,1), \quad c[2] = \chi(c_{23}) = (1,0,0,1), \\ c[3] = \chi(c_{13}) = (1,1,0,1), \\ c[4] = \chi(c_{34}) = (1,0,1,1), \quad c[5] = \chi(c_{24}) = (0,0,1,0), \\ c[6] = \chi(c_{14}) = (0,1,1,1). \end{gathered}$$

We define the operations meet ($\vee$), join ($\wedge$), difference (-) and complement ($'$) in $\{0,1\}^m$. For all vectors u, v and integers i, $1 \leq i \leq m$,

$$\begin{aligned} (u \vee v)_i &= 0 \text{ if and only if } u_i = v_i = 0, \\ (u \wedge v)_i &= 1 \text{ if and only if } u_i = v_i = 1, \\ (u - v)_i &= 1 \text{ if and only if } u_i = 1, v_i = 0, \\ (u')_i &= 1 \text{ if and only if } u_i = 0. \end{aligned}$$

Obviously, the complexity of all these operations is $O(m)$.

Let $\mathcal{S} = (U, A, \{V_a\}_{a \in A})$ be an information system such that $U = \{x_1, \ldots, x_n\}$ and $A = \{a_1, \ldots, a_m\}$. Our first algorithm tests for any $x_i, x_j \in U$ and $B \subseteq A$ whether $(x_i, x_j) \in Ind(B)$ holds or not. The Algorithm B-INDISCERNIBLE is based on Lemma 5.1.

Algorithm 5.16 B-INDISCERNIBLE
Input: An array $c[1..n(n-1)/2]$ such that for all $1 \leq i < j \leq n$, $c[k] = \chi(c_{ij})$, where $k = j(j-1)/2 - i + 1$, an m–sized vector $b = \chi(B)$, and two integers $1 \leq i, j \leq n$.
Output: "yes" if $(x_i, x_j) \in Ind(B)$ and "no" otherwise.

1. If $i = j$, then output "yes" and halt. Otherwise, go to 2.

2. If $i > j$, then $h := i,\ i := j,\ j := h$.

3. If $b \wedge c[j(j-1)/2 - i + 1] = \mathbf{0}_m$, then output "yes", otherwise output "no".

Obviously, Algorithm B-INDISCERNIBLE takes $O(m)$ time. Of course, we have assumed that the array $c[1..n(n-1)/2]$ has been computed in advance. The assumption is justified when the algorithm is applied for various choices of B and (x_i, x_j), but a fixed information system $\mathcal{S}$.

The following algorithm computes for any $x_i \in U$ and $B \subseteq A$ the equivalence class of $Ind(B)$ containing x_i, that is, the set $x_i/Ind(B)$.

Algorithm 5.17 B-CLASS
Input: An array $c[1..n(n-1)/2]$ such that for all $1 \leq i < j \leq n$, $c[k] = \chi(c_{ij})$, where $k = j(j-1)/2 - i + 1$, an m–sized vector $b = \chi(B)$, and an integer i.
Output: An n–sized vector v corresponding the set $x_i/Ind(B)$.

1. Start with $v := \mathbf{0}_n$.

2. For all $j := 1, \ldots n$, $v[j] := 1$ if $(x_i, x_j) \in Ind(B)$ holds (which can be tested with Algorithm B-INDISCERNIBLE).

3. Output v.

The complexity of Algorithm B-CLASS is $O(nm)$.

Suppose B and D are subsets of attributes. Next we present an algorithm which checks whether $Ind(B) = Ind(D)$ holds. Our Algorithm EQUIVALENT is based on Corollary 5.3.

Algorithm 5.18 EQUIVALENT
Input: An array $c[1..n(n-1)/2]$ such that for all $1 \leq i < j \leq n$, $c[k] = \chi(c_{ij})$, where $k = j(j-1)/2 - i + 1$, and two vectors $b = \chi(B)$ and $d = \chi(D)$.
Output: "yes", if $Ind(B) = Ind(D)$ holds and "no" otherwise.

1. If for all $k := 1, \ldots, n(n-1)/2$, $c[k] \wedge b = \mathbf{0}_m$ if and only if $c[k] \wedge d = \mathbf{0}_m$, then output "yes"; otherwise output "no".

The complexity of Algorithm EQUIVALENT is $O(n^2 m)$, because the size of the array c is $O(n^2)$ and the tests $c[k] \wedge b = \mathbf{0}_m$ and $c[k] \wedge c = \mathbf{0}_m$ take $O(m)$ time.

Our next algorithm returns an answer to the question whether the condition $D \to B$ holds in $\mathcal{S}$. It is based on (5.3).

Algorithm 5.19 DEPENDENCY
Input: An array $c[1..n(n-1)/2]$ such that for all $1 \leq i < j \leq n$, $c[k] = \chi(c_{ij})$, where $k = j(j-1)/2 - i + 1$, and two vectors $b = \chi(B)$ and $d = \chi(D)$.
Output: "yes", if $D \to B$ holds and "no" otherwise.

1. If for all $k := 1, \ldots, n(n-1)/2$, $c[k] \wedge b \neq \mathbf{0}_m$ implies $c[k] \wedge d \neq \mathbf{0}_m$, then output "yes"; otherwise output "no".

The complexity of this algorithm is $O(n^2m)$.

The following algorithm computes for any subset $B(\subseteq A)$ of attributes the core of B. The method is based on Proposition 5.6(b). Note that we could compute the core of B also by the condition $a \in CORE_{\mathcal{S}}(B)$ if and only if $Ind(B) \neq Ind(B - \{a\})$, and by applying Algorithm EQUIVALENT. However, this method requires $O(n^2m^2)$ time and our algorithm CORE is a little faster. Its complexity is $O(n^2m)$.

Algorithm 5.20 CORE
Input: An array $c[1..n(n-1)/2]$ such that for all $1 \leq i < j \leq n$, $c[k] = \chi(c_{ij})$, where $k = j(j-1)/2 - i + 1$, and a vector $b = \chi(B)$.
Output: A vector $v = \chi(CORE_{\mathcal{S}}(B))$.

1. Start with $v = \mathbf{0}_m$.
2. For all $k := 1, \ldots, n(n-1)/2$, if $b \wedge c[k]$ contains exactly one 1 and this is in the ith position, then $v[i] = 1$.
3. Output v.

By using Algorithm CORE it is easy to decide whether a subset $B \subseteq A$ is independent or not. Namely, $B \in IND_{\mathcal{S}}$ if and only if $B = CORE_{\mathcal{S}}(B)$. Also this test requires $O(n^2m)$ time.

It is quite easy to compute one reduct of B. We start with the set B and cancel successively its elements in a way that any set C obtained by cancelling some elements satisfies $Ind(B) = Ind(C)$. We stop this procedure if C is such that for any C_1 obtained from C by cancelling one element, we get $Ind(B) \neq Ind(C_1)$ (cf. the proof of Proposition 3.5(a)). In the sequel $v[v_i = 0]$ denotes the vector v with ith element v_i fixed to 0. The denotation $v[v_i = 1]$ is defined similarly.

Algorithm 5.21 REDUCT
Input: An array $c[1..n(n-1)/2]$ such that for all $1 \leq i < j \leq n$, $c[k] = \chi(c_{ij})$, where $k = j(j-1)/2 - i + 1$, and a vector $b = \chi(B)$.
Output: $\chi(C)$ for some $C \in RED_{\mathcal{S}}(B)$.

1. Start with $v = b$
2. For all $i := 1, \ldots, m$, let $v[i] = 0$ if for all $k = 1, \ldots, n(n-1)/2$, $c[k] \wedge b \neq \mathbf{0}_m$ implies $c[k] \wedge v[v_i = 0] \neq \mathbf{0}_m$.
3. Output v.

The complexity of this algorithm is $O(n^2m^2)$. Next we shall present a method which computes the set $RED_{\mathcal{S}}(B)$ for any $B \subseteq A$. By Proposition 5.8(a) the set of all minimal true vectors of the discernibility function $f_B^{\mathcal{S}}$ is the set of characteristic vectors of the reducts of B. Similarly, if $C \to B$ holds, then by Proposition 5.12(a) the set of all minimal true vectors of the dependency function $f_{C \to B}^{\mathcal{S}}$ equals the set of characteristic vectors of the minimal subsets D of C such that $D \to B$ holds.

Our following algorithm MF-VECTORS1 computes the set $\max F(f_B^{\mathcal{S}})$ for a discernibility function $f_B^{\mathcal{S}}$. It is based on Proposition 5.8(b).

Algorithm 5.22 MF-VECTORS1
Input: An array $c[1..n(n-1)/2]$ such that for all $1 \le i < j \le n$, $c[k] = \chi(c_{ij})$, where $k = j(j-1)/2 - i + 1$, and a vector $b = \chi(B)$.
Output: The set $\max F(f_B^{\mathcal{S}})$

1. Start with $MF := \emptyset$.
2. For all $k := 1, \ldots, n(n-1)/2$, if $c[k] \wedge b \neq \mathbf{0}_m$, then $MF := MF \cup \{(c[k] \wedge b)'\}$.
3. Delete from MF all vectors which are not maximal.
4. Output MF.

The complexity of Step 2 is $O(n^2 m)$. Because MF contains at most $n(n-1)/2$ vectors of length m after Step 2, the time need by Step 3 is $O(n^4 m)$ which is also the complexity of Algorithm MF-VECTORS1.

The following algorithm computes the set $\max F(f_{C \to B}^{\mathcal{S}})$ of a dependency function $f_{C \to B}^{\mathcal{S}}$. The method is based on Proposition 5.12.(b).

Algorithm 5.23 MF-VECTORS2
Input: An array $c[1..n(n-1)/2]$ such that for all $1 \le i < j \le n$, $c[k] = \chi(c_{ij})$, where $k = j(j-1)/2 - i + 1$, and two vectors $b = \chi(B)$ and $d = \chi(C)$ such that $C \to B$ holds.
Output: The set $\max F(f_{C \to B}^{\mathcal{S}})$

1. Start with $MF := \emptyset$.
2. For all $k := 1, \ldots, n(n-1)/2$, if $c[k] \wedge b \neq \mathbf{0}_m$, then $MF := MF \cup \{(c[k] \wedge d)'\}$.
3. Delete from MF all vectors which are not maximal.
4. Output MF.

The complexity of Algorithm MF-VECTORS2 is also $O(n^4 m)$ which the time needed by the dominating Step 3.

In what follows we shall consider the question of how to compute the set $\min T(f_B^{\mathcal{S}})$ from $\max F(f_B^{\mathcal{S}})$. These considerations have originally appeared in [1], but we have slightly altered them.

Let f be a isotone Boolean function, $MT \subseteq \min T(f)$ and $MF \subseteq \max F(f)$. That is, MT and MF, respectively, denote the partial knowledge of $\min T(f)$ and $\max F(f)$ currently in hand. In other words, MT (MF) contains all sets which at present time are surely known to be in $\min T(f)$ ($\max F(f)$). Define

$$T(MT) = \{v \mid v \ge w \text{ for some } w \in MT\} \text{ and}$$
$$F(MF) = \{v \mid v \le w \text{ for some } w \in MF\}.$$

It is clear, that $T(MT) \subseteq T(f)$ and $F(MF) \subseteq F(f)$, and $T(MT) \cap F(MF) = \emptyset$. A vector u is called *unknown* if

$$u \in \{0,1\}^m - (T(MT) \cup F(MF)),$$

since it is not known at the current stage whether u is a true vector or a false vector of f. There is no unknown vector if and only if $T(MT) \cup F(MF) = \{0,1\}^m$ holds, i.e., $MT = \min T(f)$ and $MF = \max F(f)$.

The following general algorithm computes the set $\min T(f)$ assuming that $\max F(f)$ is known. The algorithm is modified from Algorithm IDENTIFY in [1].

Algorithm 5.24 MT-VECTORS
Input: $\max F(f)$.
Output: $\min T(f)$.

1. Start with $MT := \emptyset (\subseteq \min T(f))$ and $MF := \max F(f)$.

2. Test if $T(MT) \cup F(MF) = \{0,1\}^m$ holds. If so, output MT and halt. Otherwise go to 3.

3. Find an unknown vector u (which necessarily is in $T(f)$). Then compute a minimal vector y satisfying $y \leq u$ and $y \in T(f)$. Let $MT := MT \cup \{y\}$. Return to 2.

In [1] is defined the following problem which is equivalent to Step 2.

Problem 5.25 EQ
Instance: Incomparable sets $MT, MF (\subseteq \{0,1\}^m)$ such that $T(MT) \cap F(MF) = \emptyset$.
Question: Does $T(MT) \cup F(MF) = \{0,1\}^m$ (i.e., no unknown vector) hold?

It is not known whether Problem EQ is solvable in polynomial time or not. The length of the input to Problem EQ is $m(|MT| + |MF|)$. Let us denote by T_{EQ} the time required to solve Problem EQ. We shall see that if Problem EQ is solvable in time polynomial in its input length, then finding an unknown vector in Step 3 of MT-VECTORS can also be done in polynomial time.

Let $a = (a_1, a_2, \ldots, a_k) \in \{0,1\}^k$ for some $0 \leq k < m$. We define

$$\begin{aligned}
MT[a] &= \min\{(v_{k+1}, \ldots, v_m) \mid v \in MT \text{ and } v_i \leq a_i, i = 1, 2, \ldots, k\}, \\
\min T(f)[a] &= \min\{(v_{k+1}, \ldots, v_m) \mid v \in \min T(f) \text{ and } v_i \leq a_i, i = 1, 2, \ldots, k\}, \\
MF[a] &= \max\{(v_{k+1}, \ldots, v_m) \mid v \in MF \text{ and } v_i \geq a_i, i = 1, 2, \ldots, k\}, \text{ and} \\
\max F(f)[a] &= \max\{(v_{k+1}, \ldots, v_m) \mid v \in \max F(f) \text{ and } v_i \geq a_i, i = 1, 2, \ldots, k\},
\end{aligned}$$

where v_i refers to the ith element of the vector $v = (v_1, \ldots, v_k, v_{k+1}, \ldots, v_n)$. Our following method computes the set $MT[a]$ for any $a = (a_1, \ldots, a_k)$ from the set MT. The method for the set $MF[a]$ is analogous.

Algorithm 5.26 MT[a]-VECTOR
Input: A vector $a = (a_1, \ldots, a_k)$ and the set MT.
Output: The set $MT[a]$.

1. Start with $MT_a = \emptyset$.

2. For all $v \in MT$, if for all $i = 1, \ldots, k$, $v_i \leq a_i$, then $MT_a := MT_a \cup \{(v_{k+1}, \ldots, v_m)\}$.

3. Delete from MT_a all vectors which are not minimal.

4. Output MT_a.

The complexity of Algorithm MT$[a]$-VECTOR is $O(m|MT|^2)$ which is the time needed by the step 3. Similarly, the complexity of the method MF$[a]$-VECTOR which computes $MF[a]$ from MF is $O(m|MF|^2)$.

Suppose $a = (a_1, a_2, \dots, a_k) \in \{0,1\}^k$ for some $0 \leq k < m$. Let us denote by f_a the function obtained from f by fixing variables y_i to a_i for $i = 1, \dots, k$. Then obviously $\min T(f)[a] = \min T(f_a)$ and $\max F(f)[a] = \max F(f_a)$. It is clear that if $MT \subseteq \min T(f)$ and $MF \subseteq \max F(f)$, then $MT[a] \subseteq \min T(f)[a]$ and $MF[a] \subseteq \max F(f)[a]$ for any a. Moreover, $MT[a] = \min T(f)[a]$ and $MF[a] = \max F(f)[a]$ if and only if

$$T(MT[a]) \cup F(MF[a]) = \{0,1\}^{m-k} \tag{5.6}$$

If the current $MT[a]$ and $MF[a]$ do not satisfy (5.6), then at least one of $a^0 = (a, 0)$ and $a^1 = (a, 1)$ does not satisfy (5.6) in which a is replaced by a^0 and a^1, respectively (see [1]).

This fact implies that the following algorithm, which is a corrected version of Algorithm UNKNOWN in [1], outputs an unknown vector.

Algorithm 5.27 UNKNOWN
Input: Incomparable sets $MT, MF (\subseteq \{0,1\}^m)$ such that $T(MT) \cap F(MF) = \emptyset$ and $T(MT) \cup F(MF) \neq \{0,1\}^m$, where $n \geq 2$.
Output: An unknown vector u.

1. Let $a^0 := (0)$, $a^1 := (1)$ and $k := 1$.

2. If $k < m - 1$, go to 3. If $k = m - 1$, then at least one of $M_0 := \{0,1\} - (T(MT[a^0]) \cup F(MF[a^0]))$ and $M_1 := \{0,1\} - (T(MT[a^1]) \cup F(MF[a^1]))$ is nonempty. If $M_0 \neq \emptyset$ and $b \in M_0$, let $u := (a^0, b)$ and halt. Otherwise, let $u := (a^1, b)$, where $b \in M_1$, and halt.

3. Test if $T(MT[a^0]) \cup F(MF[a^0]) = \{0,1\}^{m-k}$ holds (i.e., solve problem EQ). If "no", let $a^0 := (a^0, 0)$, $a^1 := (a^0, 1)$ and $k := k + 1$. Return to 2. Otherwise (i.e., "yes"), $a^0 := (a^1, 0)$, $a^1 := (a^1, 1)$ and $k := k + 1$. Return to 2.

Since MT$[a]$-VECTOR, MF$[a]$-VECTOR and EQ are called most m times in UNKNOWN, the complexity of Algorithm UNKNOWN is

$$O(m(m(|MT|^2 + |MF|^2) + T_{EQ}(m(|MT| + |MF|)))).$$

Next shall we consider the second half of Step 3 of MT-VECTORS. We shall present Algorithm MINIMAL, which computes a minimal true vector y from an unknown vector u (which necessary belongs to $T(f)$). The algorithm is based on the fact that for all $v \in \{0,1\}^m$ and a isotone function f, $f(v) = 1$ if and only if $v \not\leq w$ for all $w \in \max F(f)$.

Algorithm 5.28 MINIMAL
Input: A vector $u \in T(f) - T(MT)$ and $\max F(f)$.
Output: A minimal vector y such that $y \in T(f) - T(MT)$.

1. $y := u$.
2. For $i := 1, 2, \ldots, m$, let $y[i] = 0]$ if $y[y_i = 0] \not\leq v$ for all $v \in \max F(f)$.
3. Output y.

The running time of MINIMAL is clearly $O(m^2|\max F(f)|)$.

Let us consider the complete running time of Algorithm MT-VECTORS. One iteration of Steps 2 and 3 is done in $O(m(m(|MT|^2 + |MF|^2 + |\max F(f)|) + T_{EQ}(m(|MT| + |MF|))))$ time and thus the total running time of MT-VECTORS is

$$O(m|\min T(f)|(m(|\min T(f)|^2 + |\max F(f)|^2) + T_{EQ}(m(|\min T(f)| + |\max F(f)|)))).$$

Now we can present the following algorithm, which finds the reducts of given subset of an information system.

Algorithm 5.29 REDUCTS
Input: An array $c[1..n(n-1)/2]$ such that for all $1 \leq i < j \leq n$, $c[k] = \chi(c_{ij})$, where $k = j(j-1)/2 - i + 1$, and a vector $\chi(B)$.
Output: A set of vectors corresponding to $RED_{\mathcal{D}}(X)$.

1. Compute the set of vectors $\max F(f_B^{\mathcal{S}})$ with Algorithm MF-VECTORS1.
2. Compute the set of vectors $\min T(f_B^{\mathcal{S}})$ with Algorithm MT-VECTORS and output it.

We have already seen that $|\max F(f_B^{\mathcal{S}})| \leq n^2$ and $|\min T(f_B^{\mathcal{S}})| = |RED_{\mathcal{S}}(B)|$. Recall that the complexity of Step 1 is $O(n^4 m)$. Hence, the total running time of Algorithm REDUCTS is

$$O(m|RED_{\mathcal{S}}(B)|(m(|RED_{\mathcal{S}}(B)|^2 + n^4) + T_{EQ}(m(|RED_{\mathcal{S}}(B)| + n^2)))).$$

Example 5.30 As we have seen, the information system of Example 4.1 can be represented as an array $c[1..6]$, where

$$\begin{gathered} c[1] = \chi(c_{12}) = (0,1,0,1), \quad c[2] = \chi(c_{23}) = (1,0,0,1), \\ c[3] = \chi(c_{13}) = (1,1,0,1), \\ c[4] = \chi(c_{34}) = (1,0,1,1), \quad c[5] = \chi(c_{24}) = (0,0,1,0), \\ c[6] = \chi(c_{14}) = (0,1,1,1). \end{gathered}$$

We shall illustrate how Algorithm REDUCTS computes the reducts of the set $\{1,2,3,4\}$. We first have to compute the set $\max F(f_A^{\mathcal{S}})$ with Algorithm MF-VECTORS1. Obviously, $MF := \{(c[i] \wedge \chi(A))' \mid 1 \leq i \leq 6, \chi(A) \wedge c[k] \neq \mathbf{0}_m\} = \{(1,0,1,0), (0,1,1,0), (0,0,1,0), (0,1,0,0), (1,1,0,1), (1,0,0,0)\}$. The vectors $(1,1,0,1)$, $(1,0,1,0)$, and $(0,1,1,0)$ are maximal in MF, which implies $\max F(f_A^{\mathcal{S}}) = \{(1,1,0,1), (1,0,1,0), (0,1,1,0)\}$. Next we shall compute the set $\min T(f_A^{\mathcal{S}})$ with algorithm MT-VECTORS. It starts with $MT := \emptyset$ and $MF := \max F(f_A^{\mathcal{S}})$.

It is obvious that $T(MT) \cup F(MF) \neq \{0,1\}^4$, so we execute Algorithm UNKNOWN:

- $k = 1 : a^0 = (0), a^1 = (1); MT[a^0] = \emptyset, MF[a^0] = \{(1,0,1),(1,1,0)\}$.
- $k = 2 : a^0 = (0,0), a^1 = (0,1); MT[a^0] = \emptyset, MF[a^0] = \{(0,1),(1,0)\}$.
- $k = 3 : a^0 = (0,0,0), a^1 = (0,0,1); MT[a^0] = \emptyset, MF[a^0] = \{(1)\} = \{1\}$.

Now $F(MF[a^0]) = \{0,1\}$, that is, $M_0 = \emptyset$. So, we must compute the sets $MT[a^1] = \emptyset$ and $MF[a^1] = \{(0)\} = \{0\}$. Hence, $M_1 = \{1\}$, which implies $b = 1$ and $u = (0,0,1,1)$.

Because u is a minimal true vector, $MT = \{(0,0,1,1)\}$ and $MF = \{(1,1,0,1), (1,0,1,0), (0,1,1,0)\}$. Now $T(MT) \cup F(MF) \neq \{0,1\}^4$, so we shall execute UNKNOWN again:

- $k = 1 : a^0 = (0), a^1 = (1)$; $MT[a^0] = \{(0,1,1)\}, MF[a^0] = \{(1,0,1),(1,1,0)\}$.
- $k = 2 : a^0 = (1,0), a^1 = (1,1)$; $MT[a^0] = \{(1,1)\}, MF[a^0] = \{(0,1),(1,0)\}$.
- $k = 3 : a^0 = (1,1,0), a^1 = (1,1,1)$; $MT[a^0] = \emptyset$; $MF[a^0] = \{(1)\} = \{1\}$.

Because $T(MT[a^0]) \cup F(MF[a^0]) = \{0,1\}$, we have to compute the sets $MT[a^1] = \{(1)\} = \{1\}$ and $MF[a^1] = \emptyset$. So, $M_1 = \{0\}$ which implies $b = 0$ and $u = (1,1,1,0)$.

The true vector $(1,1,1,0)$ is minimal. Thus, $MT = \{(0,0,1,1), (1,1,1,0)\}$ and $MF = \{(1,1,0,1), (1,0,1,0), (0,1,1,0)\}$. Now $T(MT) \cup F(MF) = \{0,1\}^4$, which implies that $(0,0,1,1)$ and $(1,1,1,0)$ are the vectors corresponding to the reducts of A.

If the dependency $C \to B$ holds in an information system, then the following algorithm finds the set of the characteristic vectors of all minimal subsets D of C which satisfy $D \to C$.

Algorithm 5.31 MIN-DEPENDENCY
Input: An array $c[1..n(n-1)/2]$ such that for all $1 \le i < j \le n$, $c[k] = \chi(c_{ij})$, where $k = j(j-1)/2 - i + 1$, and two vectors $\chi(B)$ and $\chi(C)$ which satisfy $C \to B$ in $\mathcal{S}$.
Output: $\{\chi(D) \mid D$ is a minimal subset of C which satisfies $D \to B\}$.

1. Compute the set of vectors $\max F(f^{\mathcal{S}}_{C\to B})$ with Algorithm MF-VECTORS2.
2. Compute the set of vectors $\min T(f^{\mathcal{S}}_{C\to B})$ with Algorithm MT-VECTORS and output it.

Obviously, $|\max F(f^{\mathcal{S}}_{C\to B})| \le n^2$ and if we denote $k = |\{\chi(D) \mid D$ is a minimal subset of C which satisfies $D \to B\}|$, then the total running time of Algorithm MIN-DEPENDENCY is

$$O(mk(m(k^2+n^4) + T_{EQ}(m(k+n^2)))).$$

Example 5.32 The information system of Example 4.1 can be represented as an array $c[1..6]$, where

$$c[1] = \chi(c_{12}) = (0,1,0,1), \quad c[2] = \chi(c_{23}) = (1,0,0,1),$$
$$c[3] = \chi(c_{13}) = (1,1,0,1),$$
$$c[4] = \chi(c_{34}) = (1,0,1,1), \quad c[5] = \chi(c_{24}) = (0,0,1,0),$$
$$c[6] = \chi(c_{14}) = (0,1,1,1).$$

If we set $B = \{4\}$, then the dependency $A \to B$ holds in $\mathcal{S}$. Next we shall show how Algorithm MIN-DEPENDENCY computes the set of all minimal subset D of A which satisfy $D \to B$. First we shall compute the set $\max F(f^{\mathcal{S}}_{A\to B})$ with Algorithm MF-VECTORS2. Obivously, $MF := \{(c[i] \wedge \chi(A))' \mid 1 \le i \le 6,\ c[i] \wedge \chi(B) \ne \mathbf{0}_m\} = \{(1,0,1,0), (0,1,1,0), (0,0,1,0), (0,1,0,0), (1,0,0,0)\}$. The vectors

$(1,0,1,0)$ and $(0,1,1,0)$ are maximal in MF, which implies $\max F(f^{\mathcal{S}}_{A\to B}) = \{(1,0,1,0), (0,1,1,0)\}$. Next we shall compute the set $\min T(f^{\mathcal{S}}_{A\to B})$ with algorithm MT-VECTORS. It starts with $MT := \emptyset$ and $MF := \max F(f^{\mathcal{S}}_{A\to B})$.

It is obvious that $T(MT) \cup F(MF) \neq \{0,1\}^4$, so we have to execute Algorithm UNKNOWN:

- $k = 1 : a^0 = (0), a^1 = (1); MT[a^0] = \emptyset, MF[a^0] = \{(1,1,0)\}$.
- $k = 2 : a^0 = (0,0), a^1 = (0,1); MT[a^0] = \emptyset, MF[a^0] = \{(1,0)\}$.
- $k = 3 : a^0 = (0,0,0), a^1 = (0,0,1); MT[a^0] = \emptyset, MF[a^0] = \{(0)\} = \{0\}$.

Now $F(MF[a^0]) = \{0\}$, that is, $M_0 = \{1\}$, which implies $b = 1$ and $u = (0,0,0,1)$. Because u is minimal true vector, $MT = \{(0,0,0,1)\}$ and $MF = \{(1,0,1,0), (0,1,1,0)\}$. Clearly, $T(MT) \cup F(MF) \neq \{0,1\}^4$, so we must run UNKNOWN again:

- $k = 1 : a^0 = (0), a^1 = (1); MT[a^0] = \{(0,0,1)\}, MF[a^0] = \{(1,1,0)\}$.
- $k = 2 : a^0 = (1,0), a^1 = (1,1); MT[a^0] = \{(0,1)\}, MF[a^0] = \{(1,0)\}$.
- $k = 3 : a^0 = (1,1,0), a^1 = (1,1,1); MT[a^0] = \{(1)\} = \{1\}; MF[a^0] = \emptyset$.

Because $T(MT[a^0]) \cup F(MF[a^0]) = \{1\}$, we get $M_0 = \{0\}$ and $b = 1$. Hence, $u = (1,1,0,0)$ which is a minimal true vector. Then $MT = \{(0,0,0,1), (1,1,0,0)\}$ and $MF = \{(1,0,1,0), (0,1,1,0)\}$. Now $T(MT) \cup F(MF) = \{0,1\}^4$, which implies that $(0,0,0,1)$ and $(1,1,0,0)$ are the characteristic vectors of the subsets D of A which satisfy $D \to B$.

Chapter 6

Dependence Spaces

6.1 Congruences and closure operators on semilattices

In this section we study congruences on semilattices. Most of the results in this section appear in the literature (see [4, 14], for example), but in some cases we give new proofs. Moreover, statement (a) of Lemma 6.3 and Propositions 6.8 and 6.9 cannot be found in the mentioned sources.

In what follows, we regard a semilattice $\mathcal{P} = (P, \circ)$ also as a join-semilattice $(P, \leq)$ in which the order relations is defined by

$$a \leq b \text{ if and only if } a \circ b = b;$$

clearly, $a \circ b$ is the join of a and b in $(P, \leq)$. We say that $\mathcal{P}$ has a *zero* if there is $0 \in P$ such that $a = a \circ 0$ for all $a \in P$. Obviously, the algebra $\mathcal{P} = (P, \circ)$ has a zero if and only if the ordered set $(P, \leq)$ has a bottom element $\bot$, and in that case $0 = \bot$.

Lemma 6.1 *If $\mathcal{P} = (P, \circ)$ is a finite semilattice with a zero, then the ordered set $(P, \leq)$ is a lattice.*

Proof. Because P is finite, $\bigvee P = \bigwedge \emptyset$ exists in P and it is the greatest element. By assumption, P has a bottom element 0. Thus, $\bigwedge P = \bigvee \emptyset = 0$. If $S = \{a_1, \ldots, a_n\}$ is a nonempty subset of P, then $\bigvee S = a_1 \circ \cdots \circ a_n$. Moreover, $0 \in S^l$, which implies $S^l \neq \emptyset$. By Lemma 2.1 this yields that $\bigwedge S = \bigvee S^l$ exists for all $S \subseteq P$. □

Let $\mathcal{P} = (P, \circ)$ be a semilattice and let K be a congruence on $\mathcal{P}$, that is, for all a_1, a_2, b_1, b_2 in P, $(a_1, b_1) \in K$ and $(a_2, b_2) \in K$ imply $(a_1 \circ a_2, b_1 \circ b_2) \in K$. Let us recall that the congruence class of K containing a is denoted by a/K, and the quotient set of P modulo K is denoted by P/K. By setting

$$a/K \vee b/K = (a \circ b)/K \tag{6.1}$$

for all $a, b \in P$ we get a well-defined binary operation on P/K which is associative, commutative, and idempotent. Thus $(P/K, \vee)$ is a semilattice, the *quotient semilattice of P modulo K*. If $\leq_{P/K}$ is the the corresponding partial order, then $(P/K, \leq_{P/K})$ is a join-semilattice in which the join of any elements a/K and b/K is $a/K \vee b/K$ (which justifies our use of the symbol $\vee$).

Lemma 6.2 *Let K be a congruence on a semilattice $\mathcal{P} = (P, \circ)$.*

(a) *If $k \geq 1$ is an integer and $a_1, \ldots, a_k, b_1, \ldots, b_k$ are elements of P such that $(a_i, b_i) \in K$ for $1 \leq i \leq k$, then $(a_1 \circ \cdots \circ a_k, b_1 \circ \cdots \circ b_k) \in K$.*

(b) *If P is finite, then any K-class $B \in P/K$ contains $\bigvee B$ as its greatest element.*

(c) *If $(a, b) \in K$ and $a \leq c \leq b$, then $(b, c) \in K$*

Proof. Claim (a) follows from the definition of congruence relations by a simple induction. For (b) suppose that $B = \{a_1, \ldots, a_k\}$ is a congruence class of K. Then $(a_1, a_i) \in K$ for $1 \leq i \leq k$. By (a) and idempotency, this implies $(a_1, a_1 \circ \cdots \circ a_k) \in K$, that is, $\bigvee B \in B$. Obviously, $a_i \leq a_1 \circ \cdots \circ a_k$ for all $1 \leq i \leq k$. Hence, $\bigvee B$ is the greatest element in B. (c) Because $(a, b) \in K$ implies $(a \circ c, b \circ c) \in K$ for any $c \in P$, then $a \leq c$ and $c \leq b$ imply $(c, b) \in K$. □

Congruences on semilattices may be defined by means of closure operators. Similarly, closure operators on finite semilattices can be defined by means of congruences. In the following we shall describe these constructions.

Let $(P, \leq)$ be an ordered set. Then a function $C : P \to P$ is called a *closure operator* (see e.g. [4]), if for all $a, b \in P$,

(a) $a \leq C(a)$,

(b) $a \leq b$ implies $C(a) \leq C(b)$, and

(c) $C(C(a)) = C(a)$.

An element $a \in P$ is called *closed* if $C(a) = a$. The set of all closed elements of P is denoted by P_C.

If $(P, \leq)$ has a top element $\top$, then $\top \leq C(\top) \leq \top$ which implies that $\top \in P_C$ and it is the top element of P_C. Moreover, if P has a bottom element $\bot$, then $\bot \leq a$ for all $a \in P$, which implies $C(\bot) \leq C(a)$ for all $a \in P$. Thus, P_C has a bottom element $C(\bot)$. We extent C to subsets of P in the natural way: for $S \subseteq P$, $C(S) = \{C(a) \mid a \in S\}$.

Lemma 6.3 *If $(P, \leq)$ is a complete lattice and $C : P \to P$ is a closure operator, then the following facts hold.*

(a) $C(\bigvee S) = C(\bigvee C(S))$ *for all* $S \subseteq P$.

(b) $P_C = \{C(a) \mid a \in P\}$.

(c) $\bigwedge_P S \in P_C$ *for all* $S \subseteq P_C$.

(d) $C(a) = \bigwedge_P \{b \in P_C \mid a \leq b\}$.

(e) $(P_C, \leq)$ *is a complete lattice such that for every subset S of P_C,*

$$\begin{aligned} \textstyle\bigwedge_{P_C} S &= \textstyle\bigwedge_P S, \\ \textstyle\bigvee_{P_C} S &= C(\textstyle\bigvee_P S). \end{aligned}$$

Proof. (a) Suppose $S \subseteq P$. For all $a \in S$, $C(a) \leq C(\bigvee S)$ since $a \leq \bigvee S$. Hence, $\bigvee C(S) \leq C(\bigvee S)$ and hence $C(\bigvee C(S)) \leq C(C(\bigvee S)) = C(\bigvee S)$. On the other hand, $a \leq C(a) \leq \bigvee\{C(a) \mid a \in S\} = \bigvee C(S)$ for all $a \in S$, which implies $\bigvee S \leq \bigvee C(S)$ and $C(\bigvee S) \leq C(\bigvee C(S))$.

(b) If $a \in P$ is closed, then $C(a) = a$, that is, $a \in \{C(b) \mid b \in P\}$. Conversely, if $a \in \{C(b) \mid b \in P\}$, then $a = C(b)$ for some $b \in P$. Obviously, $C(a) = C(C(b)) = C(b) = a$, i.e., a is closed.

(c) Suppose $S \subseteq P_C$. Then $\bigwedge_P S \leq C(\bigwedge_P S)$. It is clear that $\bigwedge_P S \leq a$ for all $a \in S$, which implies $C(\bigwedge_P S) \leq C(a) = a$ for all $a \in S$. Hence, $C(\bigwedge_P S)$ is a lower bound for S and thus $C(\bigwedge_P S) \leq \bigwedge_P S$.

(d) Obviously, $C(a) \leq \bigwedge_P \{b \in P_C \mid a \leq b\}$. Because $C(a) \in P_C$, this implies $C(a) \in \{b \in P_C \mid a \leq b\}$. Hence, $\bigwedge_P \{b \in P_C \mid a \leq b\} \leq C(a)$.

(e) We have seen that P_C has the top element $\top$ and the bottom element $C(\bot)$. Hence, $\bigwedge_{P_C} \emptyset = \top = \bigwedge_P \emptyset$, $\bigwedge_{P_C} P_C = C(\bot) = \bigwedge_P P_C$, $\bigvee_{P_C} \emptyset = C(\bot) = C(\bigvee_P \emptyset)$, and $\bigvee_{P_C} P_C = \top = C(\top) = C(\bigvee_P P_C)$. If S is a nonempty subset of P_C, then $\bigwedge_P S \in P_C$ by (c) and hence $\bigwedge_{P_C} S = \bigwedge_P S$. By Lemma 2.1,

$$\begin{aligned}
\bigvee_{P_C} S &= \bigwedge_{P_C} S^u \\
&= \bigwedge_P \{b \in P_C \mid a \leq b \text{ for all } a \in S\} \\
&= \bigwedge_P \{b \in P_C \mid \bigvee_P S \leq b\} \\
&= C(\bigvee_P S).
\end{aligned}$$

□

Note that Lemma 6.3(c) holds for any ordered set $(P, \leq)$.

If $\mathcal{P} = (P, \circ)$ is a finite semilattice and K is a congruence on $\mathcal{P}$, then by Lemma 6.2(b) the block a/K has the greatest element $\bigvee a/K$ for all $a \in P$. We define the following function.

$$C_K : P \to P, \quad a \mapsto \bigvee a/K.$$

Then the following proposition holds.

Proposition 6.4 *If $\mathcal{P} = (P, \circ)$ is a finite semilattice and K is a congruence on $\mathcal{P}$, then the mapping C_K is a closure operator. Moreover,* $\ker C_K = K$.

Proof. We show that C_K satisfies conditions (a)–(c) in the definition of closure operators. (a) The fact $a \in a/K$ implies $a \leq \bigvee a/K = C_K(a)$. (b) Assume $a \leq b$. Then $(a, C_K(a)) \in K$ and $(b, C_K(b)) \in K$ imply $(a \circ b, C_K(a) \circ C_K(b)) = (b, C_K(a) \circ C_K(b)) \in K$. Hence, $C_K(a) \circ C_K(b) \leq C_K(b)$. Since $x \circ y \geq y$ for any $x, y \in P$, this implies that $C_K(a) \circ C_K(b) = C_K(b)$, i.e., $C_K(a) \leq C_K(b)$. (c) $C_K(C_K(a)) = \bigvee C_K(a)/K = \bigvee a/K = C_K(a)$.

Suppose $(a, b) \in K$. Then $(a, C_K(a)) \in K$ and $(b, C_K(b)) \in K$ imply $(b, C_K(a)) \in K$ and $(a, C_K(b)) \in K$. Hence, $C_K(a) \leq C_K(b)$ and $C_K(b) \leq C_K(a)$, i.e., $C_K(a) = C_K(b)$. On the other hand, assume $C_K(a) = C_K(b)$. Then the facts $(a, C_K(a)) \in K$ and $(C_K(b), b) \in K$ imply $(a, b) \in K$. □

If K is a congruence on a finite semilattice $(P, \circ)$, then we denote the set $P_{(C_K)}$ simply by P_K. By the previous proposition, every congruence on a finite semilattice defines a closure operator. In what follows we shall see how every closure operator defines a congruence relation.

Proposition 6.5 *If $\mathcal{P} = (P, \circ)$ is a finite semilattice with a zero and $C : P \to P$ is a closure operator, then the following facts hold.*

(a) *$(P_C, \leq)$ is a lattice.*

(b) *The mapping C is a homomorphism from $(P, \circ)$ onto $(P_C, \vee_{P_C})$.*

(c) *The* $\ker C$ *is a congruence on $\mathcal{P}$.*

Proof. Statement (a) follows from Lemmas 6.1 and 6.3(e). (b) By Lemma 6.3, $C(a \circ b) = C(C(a) \circ C(b)) = C(a) \vee_{P_C} C(b)$. Hence, C is a homomorphism. Moreover, $P_C = \{C(a) \mid a \in P\}$ which implies that C is onto. That (c) follows from (b) is a well-known fact of general algebra. □

In the sequel we denote $\ker C$ by K_C.

We have shown that if $\mathcal{P} = (P, \circ)$ is a finite semilattice with a zero, then every congruence K on $\mathcal{P}$ defines a closure operator $C_K : P \to P$, and every closure operator $C : P \to P$ defines a congruence K_C on $\mathcal{P}$. Moreover, the following lemma holds.

Lemma 6.6 *Let $\mathcal{P} = (P, \circ)$ be a finite semilattice with a zero.*

(a) *If K is a congruence on $\mathcal{P}$, then $K = K_{(C_K)}$.*

(b) *If $C : P \to P$ is a closure operator, then $C = C_{(K_C)}$.*

Proof. (a) If K is a congruence on $\mathcal{P}$, then for all $a, b \in P$, $(a, b) \in K \Leftrightarrow C_K(a) = C_K(b) \Leftrightarrow (a, b) \in K_{(C_K)}$.

(b) If $C : P \to P$ is a closure operator, then for all $a, b \in P$, $C(a) = C(b) \Leftrightarrow (a, b) \in K_C \Leftrightarrow C_{(K_C)}(a) = C_{(K_C)}(b)$. □

By Lemma 6.6 we can write the following proposition.

Proposition 6.7 *If $\mathcal{P} = (P, \circ)$ is a finite semilattice with a zero, then the mappings $C \mapsto K_C$ and $K \mapsto C_K$ form a pair of mutually inverse bijections between the set of all closure operators $C : P \to P$ and the set of all congruences on $\mathcal{P}$.* □

If $\mathcal{P} = (P, \circ)$ is a finite semilattice with 0 and K is a congruence on $\mathcal{P}$, then the quotient semilattice has a least element $0/K$, and hence it is a lattice by Lemma 6.1. Therefore, we can write the following proposition.

Proposition 6.8 *Let $\mathcal{P} = (P, \circ)$ be a finite semilattice with a zero and let K be a congruence on $\mathcal{P}$. If we set*

$$\begin{aligned} a/K \vee b/K &= (a \circ b)/K, \quad \textit{and} \\ a/K \wedge b/K &= (C_K(a) \wedge_P C_K(b))/K, \end{aligned}$$

then the algebra $(P/K, \vee, \wedge)$ is a lattice.

Proof. We have already seen that the well-defined binary operation $\vee$ on P/K is associative, commutative, and idempotent. The operation $\wedge$ is also well-defined on P/K, and clearly it is commutative and idempotent. For all $a, b, c \in P$,

$$\begin{gathered} a/K \wedge (b/K \wedge c/K) = a/K \wedge (C_K(b) \wedge_P C_K(c))/K \\ = (C_K(a) \wedge_P C_K(C_K(b) \wedge_{P_K} C_K(c)))/K = (C_K(a) \wedge_P C_K(b) \wedge_{P_K} C_K(c))/K \\ = (C_K(a) \wedge_{P_K} C_K(b) \wedge_P C_K(c))/K = (C_K(C_K(a) \wedge_P C_K(b)) \wedge_P C_K(c))/K \\ = (C_K(a) \wedge_P C_K(b))/K \wedge C/K = (a/K \wedge b/K) \wedge c/K. \end{gathered}$$

Hence, $\wedge$ is associative. Next we show that the absorption identities (L4) and (L4)$^\partial$ hold.

$$a/K \vee (a/K \wedge b/K) = C_K(a) \vee (C_K(a) \wedge_P C_K(b))/K$$
$$= (C_K(a) \circ (C_K(a) \wedge_P C_K(b))/K = C_K(a)/K = a/K.$$

Similarly,

$$a/K \wedge (a/K \vee B/K) = a/K \wedge (a \circ b)/K$$
$$= (C_K(a) \wedge_P C_K(a \circ b)/K = C_K(a)/K = a/K.$$

Hence, the algebra $(P/K, \vee, \wedge)$ is a lattice. □

If $\mathcal{P} = (P, \circ)$ is a finite semilattice with a zero, then for every closure operator $C : P \to P$, the set $(P_C, \leq)$ is a lattice by Proposition 6.5(a). In particular, if K is a congruence on $\mathcal{P}$, then $(P_K, \vee_{P_K}, \wedge_{P_K})$, where $a \vee_{P_K} b = (a \circ b)/K$ and $a \wedge_{P_K} b = a \wedge_P b$, is a lattice. Next we shall show that the lattices $(P/K, \vee, \wedge)$ and $(P_K, \vee_{P_K}, \wedge_{P_K})$ are isomorphic.

Proposition 6.9 *If $\mathcal{P} = (P, \circ)$ is a finite semilattice with a zero and K is a congruence on $\mathcal{P}$, then*

$$\begin{aligned} \varphi : P/K &\to P_K, \\ a/K &\mapsto C_K(a) \end{aligned}$$

defines an isomorphism between the lattices $(P/K, \vee, \wedge)$ and $(P_K, \vee_{P_K}, \wedge_{P_K})$.

Proof. By Proposition 6.5(b), the closure operator C_K is a homomorphism from $(P, \circ)$ onto $(P_K, \vee_{P_K})$. Because by Proposition 6.4, $\ker C_K = K$, then the function $\varphi : P/K \to P_K$, given by $\varphi(a/K) = C_K(a)$, is an isomorphism between $(P/K, \vee)$ and $(P_K, \vee_{P_K})$ by Homomorphism Theorem. Because $(P/K, \vee, \wedge)$ and $(P_K, \vee_{P_K}, \wedge_{P_K})$ are lattices, the claim follows from Proposition 2.14. □

6.2 Congruences and dense sets of semilattices

In the previous section we saw how every congruence on a finite semilattice defines a closure operator, and vice versa. In this section we show that every subset of a semilattice defines a congruence on that same semilattice.

The following binary relation is defined in [14]. Suppose $\mathcal{P} = (P, \circ)$ is a semilattice and $T \subseteq P$. Then we define a binary relation K_T on P by

$$K_T = \{(b, c) \in P^2 \mid \text{ for all } a \in T,\ b \leq a \text{ if and only if } c \leq a\}.$$

Then the following lemma holds.

Lemma 6.10 *Let $\mathcal{P} = (P, \circ)$ be a semilattice. Then for all $T \subseteq P$, the relation K_T is a congruence on $\mathcal{P}$.*

Proof. It is obvious that K_T is an equivalence relation. Suppose $(b_1, c_1) \in K_T$, $(b_2, c_2) \in K_T$ and $a \in T$. If $b_1 \circ b_2 \leq a$, then $b_1, b_2 \leq a$, which implies $c_1, c_2 \leq a$, from which we deduce $c_1 \circ c_2 \leq a$. Similarly, $c_1 \circ c_2 \leq a$ implies $b_1 \circ b_2 \leq a$. Thus $(b_1 \circ b_2, c_1 \circ c_2) \in K_T$. □

The following definition of dense sets can also be found from [14] in different form. Let $\mathcal{P} = (P, \circ)$ be a semilattice and assume K is a congruence on $\mathcal{P}$. Then a subset T of P is called *dense with respect to* K if $K = K_T$.

If $\mathcal{P} = (P, \circ)$ is a finite semilattice and K is a congruence on $\mathcal{P}$, then we can denote as in the previous section the closure operator corresponding K by C_K, and the set of closed elements corresponding to C_K is by P_K. We have the following result.

Proposition 6.11 *If $\mathcal{P} = (P, \circ)$ is a finite semilattice and K is a congruence of $\mathcal{P}$, then the following facts hold.*

(a) $K = K_{(P_K)}$.

(b) *If T is dense with respect to K, then $T \subseteq P_K$.*

Proof. (a) If $(b, c) \in K$, then $C_K(b) = C_K(c)$. Let $a \in P_K$. If $b \leq a$, then $c \leq C_K(c) = C_K(b) \leq C_K(a) = a$. Similarly, $c \leq a$ implies $b \leq a$. Hence, $(b, c) \in K_{(P_K)}$. Conversely, if $(b, c) \in K_{(P_K)}$, then for all $a \in P_K$, the conditions $b \leq a$ and $c \leq a$ are equivalent. Because $C_K(b) \in P_K$ and $b \leq C_K(b)$, then $c \leq C_K(b)$ which implies $C_K(c) \leq C_K(C_K(b)) = C_K(b)$. Similarly, we can show that $C_K(b) \leq C_K(c)$. Hence, $C_K(b) = C_K(c)$, i.e., $(b, c) \in K$.

(b) Suppose T is a dense with respect to K and let a be an arbitrary element of T. If $a \notin P_K$, then $a < C_K(a)$. Because $(a, C_K(a)) \in K_T$ and $a \leq a (\in T)$, we get $C_K(a) \leq a$, a contradiction! Hence, $a \in P_K$. □

We have shown that if $\mathcal{P} = (P, \circ)$ is a finite semilattice, then every subset of P defines a congruence on $\mathcal{P}$. Similarly, every congruence K on $\mathcal{P}$ defines a family of dense sets $\mathcal{T}$ such that $K = K_T$ for all $T \in \mathcal{T}$. Moreover, the set P_K is the greatest dense set. Next we try to find the least dense set.

We know that if $\mathcal{P} = (P, \circ)$ is a finite semilattice with a zero and K is a congruence on $\mathcal{P}$, then the algebra $(P_K, \vee_{P_K}, \wedge_{P_K})$, where $a \vee_{P_K} b = C_K(a \circ b)$ and $a \wedge_{P_K} b = a \wedge_P b$, is a lattice. If we denote by $\mathcal{M}(P_K)$ the set of meet-irreducible elements $a \neq 1$ of P_K, then the following proposition holds.

Proposition 6.12 *Let $\mathcal{P} = (P, \circ)$ be a finite semilattice with a zero and let K be a congruence on $\mathcal{P}$. Then $\mathcal{M}(P_K)$ is the least dense set with respect to K.*

Proof. First, we shall show that $\mathcal{M}(P_K)$ is dense. The fact $\mathcal{M}(P_K) \subseteq P_K$, obviously implies $K = K_{(P_K)} \subseteq K_{\mathcal{M}(P_K)}$. Suppose $(b, c) \in K_{\mathcal{M}(P_K)}$ and $(b, c) \notin K_{(P_K)}$ for some $b, c \in P$. Then there exists $a \in P_K - \mathcal{M}(P_K)$ such that either (i) $b \leq a$ and $c \not\leq a$ or (ii) $b \not\leq a$ and $c \leq a$ holds. Let us denote $S = \{x \in \mathcal{M}(P_K) \mid a \leq x\}$. Because P is finite, the lattice $(P_K, \leq)$ satisfies the ACC. Then by Lemma 2.13 and Lemma 6.3(e), $a = \bigwedge_{P_K} S = \bigwedge_P S$. Let us consider the case (i). The condition $b \leq a$ implies that $b \leq x$ for all $x \in S$. Because $S \subseteq \mathcal{M}(P_K)$, $c \leq x$ for all $x \in S$. Hence, c is a lower bound for S, which implies $c \leq \bigwedge_P S = a$, a contradiction! Similarly, the case (ii) leads to contradiction. Thus $(b, c) \in K_{(P_K)}$, which implies that the set $\mathcal{M}(P_K)$ is dense.

Secondly, we shall show that $\mathcal{M}(P_K)$ is the least dense set. Assume $\mathcal{M}(P_K) - T \neq \emptyset$ for some dense set T. This implies that there exists $a \in \mathcal{M}(P_K) - T$. Because $a \in \mathcal{M}(P_K)$ and P_K is finite, there exists exactly one $b \in P_K$ which satisfies $a \prec b$ in P_K. Clearly, for all $x \in T$, $b \leq x$ implies $a \leq x$. Suppose there exists $x \in T$ such that

$a \le x$ and $b \not\le x$. Because $T \subseteq P_K$, $a \le x \wedge_{P_K} b < b$. The condition $a \prec b$ (in P_K) implies $a = x \wedge_{P_K} b$. Because a is a meet-irreducible element of P_K, $a = x$ or $a = b$. Obviously both of these equations lead to a contradiction. Hence, for all $x \in T$, $a \le x$ implies $b \le x$. Thus $(a, b) \in K_{(P_K)} = K$, i.e., $C_K(a) = C_K(b)$. Because $a, b \in P_K$, we get $a = C_K(a) = C_K(b) = b$, a contradiction! Thus, $\mathcal{M}(P_K) - T = \emptyset$ for all dense sets T. □

By Propositions 6.11 and 6.12 we can now give the following characterization of dense sets.

Proposition 6.13 *If $\mathcal{P} = (P, \circ)$ be a finite semilattice with a zero and assume K is a congruence on $\mathcal{P}$, then $T(\subseteq P)$ is dense with respect to K if and only if $\mathcal{M}(P_K) \subseteq T \subseteq P_K$.* □

Our two following results show how to compute the value $C_K(a)$ for any $a \in P$.

Lemma 6.14 *If $\mathcal{P} = (P, \circ)$ be a finite semilattice with a zero and assume K is a congruence on $\mathcal{P}$, then*

$$C_K(a) = \bigwedge\nolimits_P \{b \in \mathcal{M}(P_K) \mid a \le b\}.$$

Proof. Because $b \in P_K$, $a \le b$ if and only if $C_K(a) \le b$, and hence the equation follows directly from $C_K(a) = \bigwedge_P \{b \in \mathcal{M}(P_K) \mid C_K(a) \le b\}$. □

Proposition 6.15 *Let $\mathcal{P} = (P, \circ)$ be a finite semilattice with a zero and assume K is a congruence on $\mathcal{P}$. If T is a dense subset of P, then*

$$C_K(a) = \bigwedge\nolimits_P \{b \in T \mid a \le b\}.$$

Proof. By Lemma 6.3(d) and Proposition 6.4,

$$C_K(a) = \bigwedge\nolimits_P \{b \in P_K \mid a \le b\}$$

and by Lemma 6.14,

$$C_K(a) = \bigwedge\nolimits_P \{b \in \mathcal{M}(P_K) \mid a \le b\}.$$

Proposition 6.13 implies that if T is a dense subset of P, $\mathcal{M}(P_K) \subseteq T \subseteq P_K$ holds. Hence for all $a \in P$,

$$\{b \in \mathcal{M}(P_K) \mid a \le b\} \subseteq \{b \in T \mid a \le b\} \subseteq \{b \in P_K \mid a \le b\},$$

which implies by Lemma 2.3,

$$C_K(a) = \bigwedge\nolimits_P \{b \in P_K \mid a \le b\} \le \bigwedge\nolimits_P \{b \in T \mid a \le b\} \le \bigwedge\nolimits_P \{b \in \mathcal{M}(P_K) \mid a \le b\} = C_K(a),$$

that is, $C_K(a) = \bigwedge_P \{b \in T \mid a \le b\}$. □

6.3 Closure operators and dense sets of dependence spaces

We recall Novotný's and Pawlak's definition of dependence spaces (see [12], for example). The considerations of this section are mainly special cases of the results presented in Section 6.1. If A is a nonempty set, then the algebra $(\wp(A), \cup)$ is a semilattice which has $\emptyset$ as the zero element. Since $B \subseteq C$ if and only if $B \cup C = C$ for all $B, C \in \wp(A)$, the corresponding join-semilattice is $(\wp(A), \subseteq)$. If A is a finite nonempty set and K a congruence on the semilattice $(\wp(A), \cup)$, then the ordered pair $\mathcal{D} = (A, K)$ is said to be a *dependence space*.

Let $\mathcal{D} = (A, K)$ be a dependence space. The operation of the corresponding quotient semilattice $(\wp(A)/K, \vee)$ is defined by $B/K \vee C/K = (B \cup C)/K$. Since for all $B, C \in \wp(A)$, $B/K \leq C/K$ if and only if $B/K \vee C/K = C/K$, the partial order is given by the condition

$$B/K \leq C/K \text{ if and only if } (B \cup C)/K = C/K. \tag{6.2}$$

For a dependence space $\mathcal{D} = (A, K)$ a mapping $\mathcal{C}_{\mathcal{D}} : \wp(A) \to \wp(A)$ is defined by

$$\mathcal{C}_{\mathcal{D}}(B) = \bigcup B/K$$

for all $B \subseteq A$. Recalling the finiteness of the semilattice $(\wp(A), \cup)$, it is obvious by Lemma 6.2(b) that for every $B (\subseteq A)$, the block B/K contains $\mathcal{C}_{\mathcal{D}}(B)$ which is its greatest element. By Lemma 6.4 it is clear that $\mathcal{C}_{\mathcal{D}}$ is a closure operator and

$$(B, C) \in K \text{ if and only if } \mathcal{C}_{\mathcal{D}}(B) = \mathcal{C}_{\mathcal{D}}(C) \tag{6.3}$$

for all $B, C \in \wp(A)$. From conditions (6.2) and (6.3) and Lemma 2.6(b) it follows, that we can determine $\mathcal{C}_{\mathcal{D}}(B)$ for every $B \subseteq A$ by the rule

$$\mathcal{C}_{\mathcal{D}}(B) = \{a \in A \mid \{a\}/K \leq B/K\}.$$

We have seen how every dependence space $\mathcal{D} = (A, K)$ defines a closure operator $\mathcal{C}_{\mathcal{D}}$. The following Lemma is obvious by Lemma 6.5(c).

Lemma 6.16 *Let $\mathcal{C}$ be a closure operator on a finite set A. If we define a binary relation $K_{\mathcal{C}}$ on $\wp(A)$ by setting*

$$(B, C) \in K_{\mathcal{C}} \textit{ if and only if } \mathcal{C}(B) = \mathcal{C}(C)$$

for all $B, C \in \wp(A)$, then $\mathcal{D}_{\mathcal{C}} = (A, K_{\mathcal{C}})$ is a dependence space. □

Thus, every closure operator $\mathcal{C} : \wp(A) \to \wp(A)$ defines a dependence space $\mathcal{D}_{\mathcal{C}} = (A, K_{\mathcal{C}})$. By Lemma 6.6 we can write the following lemma.

Lemma 6.17 *Let A be a finite set.*

(a) *If $\mathcal{D} = (A, K)$ is a dependence space, then $K = K_{(\mathcal{C}_{\mathcal{D}})}$.*

(b) *If $\mathcal{C}$ is a closure operator on A, then $\mathcal{C} = \mathcal{C}_{(\mathcal{D}_{\mathcal{C}})}$.* □

The following proposition is clear by Proposition 6.7.

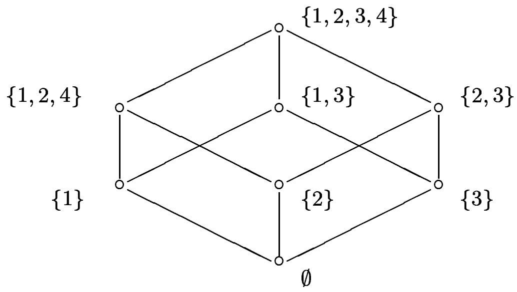

Figure 5

Proposition 6.18 *For any finite set A, the mappings $\mathcal{C} \mapsto \mathcal{D}_{\mathcal{C}}$ and $\mathcal{D} \mapsto \mathcal{C}_{\mathcal{D}}$ form a pair of mutually inverse bijections between the sets of all closure operators $\mathcal{C} : \wp(A) \to \wp(A)$ and the set of all dependence spaces $\mathcal{D} = (A, K)$.* □

It is obvious that every dependence space $\mathcal{D} = (A, K)$ defines a closure system $\mathcal{L}_{(\mathcal{C}_{\mathcal{D}})}$ (denoted shortly by $\mathcal{L}_{\mathcal{D}}$) on A. Clearly,

$$\mathcal{L}_{\mathcal{D}} = \{B \subseteq A \mid B = \bigcup B/K\}.$$

Note that $\mathcal{L}_{\mathcal{D}}$ consists of the greatest elements with respect to inclusion of the K-classes. The following proposition is obvious by Propositions 6.8 and 6.9.

Proposition 6.19 *Let $\mathcal{D} = (A, K)$ be a dependence space.*

(a) *If we set*

$$B/K \vee C/K = (B \cup C)/K \text{ and}$$
$$B/K \wedge C/K = (\mathcal{C}_{\mathcal{D}}(B) \cap \mathcal{C}_{\mathcal{D}}(C))/K$$

for all $B, C \in \wp(A)$, then the algebra $(\wp(A)/K, \vee, \wedge)$ is a lattice.

(b) *The mapping $\varphi : \wp(A)/K \to \mathcal{L}_{\mathcal{D}}$, $C/K \mapsto \mathcal{C}_{\mathcal{D}}(C)$, defines an isomorphism between the lattices $(\wp(A)/K, \vee, \wedge)$ and $(\mathcal{L}_{\mathcal{D}}, \vee_{\mathcal{L}_{\mathcal{D}}}, \wedge_{\mathcal{L}_{\mathcal{D}}})$.* □

By Theorem 2.14 it is clear that for all $B, C (\subseteq A)$,

(6.4) $$\mathcal{C}_{\mathcal{D}}(B) \subseteq \mathcal{C}_{\mathcal{D}}(C) \text{ if and only if } B/K \leq C/K.$$

Example 6.20 Let $A = \{1, 2, 3, 4\}$ and K be the congruence on $(\wp(A), \cup)$ whose congruence classes are $\{\emptyset\}$, $\{\{1\}\}$, $\{\{2\}\}$, $\{\{3\}\}$, $\{\{4\}$, $\{1, 2\}$, $\{1, 4\}$, $\{2, 4\}$, $\{1, 2, 4\}\}$, $\{\{1, 3\}\}$, $\{\{2, 3\}\}$ and $\{\{3, 4\}$, $\{1, 2, 3\}$, $\{1, 3, 4\}$, $\{2, 3, 4\}$, $\{1, 2, 3, 4\}\}$. The closure lattice $(\mathcal{L}_{\mathcal{D}}, \subseteq)$ corresponding dependence space $\mathcal{D} = (A, K)$ is presented in Figure 5. Moreover, $\mathcal{M}(\mathcal{L}_{\mathcal{D}}) = \{\{1, 2, 4\}, \{1, 3\}, \{2, 3\}\}$.

6.4 Independent sets and reducts

We shall review some notions and basic results concerning dependence spaces. Lemmas 6.21 and 6.23 can be found also in [10, 12, 14]. Earlier versions of other results in this section are presented in [8] where they are formulated by the means of closure systems. Here we present similar considerations by applying dense sets. Our main result in this section gives a new characterization to the reducts of a given subset of a dependence space.

Let $\mathcal{D} = (A, K)$ be a dependence space. A subset $B(\subseteq A)$ is called *independent*, if B is minimal with respect to inclusion in its K-class; otherwise it is called *dependent*. We denote the set of all independent subsets of $\mathcal{D}$ by $IND_{\mathcal{D}}$. The next lemma characterizes the independent subsets of a dependence space. Moreover, it shows that every subset of an independent set is independent.

Lemma 6.21 *If $\mathcal{D} = (A, K)$ is a dependence space and $B \subseteq A$, then the following conditions hold.*

(a) *$B \in IND_{\mathcal{D}}$ if and only if $(B, B - \{a\}) \notin K$ for all $a \in B$.*

(b) *If $B \in IND_{\mathcal{D}}$ and $C \subseteq B$, then $C \in IND_{\mathcal{D}}$.*

Proof. If $B \in IND_{\mathcal{D}}$, then obviously $(B, B - \{a\}) \notin K$ for all $a \in B$. Conversely, suppose $(B, B - \{a\}) \notin K$ holds for all $a \in B$. If $B \notin IND_{\mathcal{D}}$, then there exists $C \subset B$ which satisfies $(B, C) \in K$. If $a \in B - C$, then $C \subseteq B - \{a\} \subseteq B$, which implies by Lemma 6.2(c) that $(B, B - \{a\}) \in K$, a contradiction!

(b) Suppose $B(\subseteq A)$ is independent and $C \subseteq B$. If $C \notin IND_{\mathcal{D}}$, then there exists $a \in C$ such that $(C, C - \{a\}) \in K$. This implies $(B, B - \{a\}) = (C \cup (B - C), (C - \{a\}) \cup (B - C)) \in K$, a contradiction! ◻

Remark 6.22 Let us note that this notion of independence is actually equivalent to a general notion of independence with respect to a closure operator. Let $\mathcal{C}$ be a closure operator on A. The set $B \subseteq A$ is said to be *$\mathcal{C}$-independent* if $a \notin \mathcal{C}(B - \{a\})$ for every $a \in B$ (cf. [6], for example). If $\mathcal{D} = (A, K)$ is a dependence space and $B \subseteq A$, then by (6.3) and Lemma 2.6(b), $(B, B - \{a\}) \notin K \Leftrightarrow \mathcal{C}_{\mathcal{D}}(B) \neq \mathcal{C}_{\mathcal{D}}(B - \{a\}) \Leftrightarrow a \notin \mathcal{C}_{\mathcal{D}}(B - \{a\})$ for all $a \in B$. Hence, for all $B \subseteq A$, $B \in IND_{\mathcal{D}}$ if and only if B is $\mathcal{C}_{\mathcal{D}}$-independent.

As we have already shown, the notion of dependence spaces could equivalently be defined as a pair $\mathcal{D} = (A, \mathcal{C})$, where $\mathcal{C}$ is a closure operator on a finite set A. By our remark, the set $IND_{\mathcal{D}}$ can be defined in this structure by the means of a general notion of independence known in the literature.

For any $B(\subseteq A)$, a set $C(\subseteq A)$ is a *reduct* of B if $C \subseteq B$, $(B, C) \in K$, and $C \in IND_{\mathcal{D}}$. The set of all reducts of B will be denoted by $RED_{\mathcal{D}}(B)$. In the other words, a subset $C(\subseteq B)$ is a reduct of B, if C is minimal in the block B/K with respect to the inclusion relation. Because A is finite, it is obvious that the ordered set $(\wp(A), \subseteq)$ is finite. Hence, it satisfies the DCC. Because $\{C \subseteq B \mid C \in B/K\}$ is a nonempty subset of $\wp(A)$, by the dual of Lemma 2.4 this implies that $\{C \subseteq B \mid C \in B/K\}$ has a minimal element i.e., a reduct of B.

An element $a \in B$ is said to be *indispensable* for B if $(B, B - \{a\}) \notin K$. The set of all indispensable elements forms the *core* of B, which is denoted by $CORE_{\mathcal{D}}(B)$.

Lemma 6.23 *If $\mathcal{D} = (A, K)$ is a dependence space and $B \subseteq A$, then for all $B \subseteq A$,*

$$CORE_{\mathcal{D}}(B) = \bigcap RED_{\mathcal{D}}(B).$$

Proof. Assume $a \in CORE_{\mathcal{D}}(B)$ and $a \notin C$ for some $C \in RED_{\mathcal{D}}(B)$. Hence, $C \subseteq B - \{a\} \subseteq B$ and $(C, B) \in K$, which implies by Theorem 6.2(c) that $(B, B - \{a\}) \in K$, a contradiction!

Conversely, suppose $a \in \bigcap RED_{\mathcal{D}}(B)$ and $(B, B - \{a\}) \in K$. Because $(\wp(A), \subseteq)$ satisfies the DCC, then by the dual of Lemma 2.4, a nonempty subset $\{C \subseteq B - \{a\} \mid (B, C) \in K\}$ of $\wp(A)$ has a minimal element C. Trivially, $C \in RED_{\mathcal{D}}(B)$. Because $a \notin C$ this implies $a \notin \bigcap RED_{\mathcal{D}}(B)$, a contradiction! □

Finding all reducts of a given set is known as the reduction problem. In what follows we shall study this basic problem closely. By definition, a dependence space is a pair $\mathcal{D} = (A, K)$ in which A is a finite nonempty set and K is a congruence on the semilattice $(\wp(A), \cup)$. Moreover, $\wp(A)$ has a zero element $\emptyset$. We say that a subset $\mathcal{T} \subseteq \wp(A)$ is *dense* in a dependence space $\mathcal{D}$ if $\mathcal{T}$ is a dense with respect to K subset of $\wp(A)$ in the sense of Section 6.2, that is,

$$K = K_{\mathcal{T}} = \{(B, C) \in \wp(A)^2 \mid \text{ for all } L \in \mathcal{T}, B \subseteq L \text{ if and only if } C \subseteq L\}.$$

It is clear that $\mathcal{T}$ is dense if and only if $\mathcal{M}(\mathcal{L}_{\mathcal{D}}) \subseteq \mathcal{T} \subseteq \mathcal{L}_{\mathcal{D}}$.

By Proposition 6.15 we can write the following result.

Proposition 6.24 *Let $\mathcal{D} = (A, K)$ be a dependence space. If a subset $\mathcal{T} (\subseteq \wp(\mathcal{A}))$ is dense, then*

$$\mathcal{C}_{\mathcal{D}}(B) = \bigcap \{L \in \mathcal{T} \mid B \subseteq L\}$$

for all $B \subseteq A$. □

Lemma 6.25 *Let $\mathcal{D} = (A, K)$ be a dependence space. If a subset $\mathcal{T} (\subseteq \wp(\mathcal{A}))$ is dense, then the following conditions are equivalent for all $B, C \subseteq A$.*

(a) $\mathcal{C}_{\mathcal{D}}(B) \subseteq \mathcal{C}_{\mathcal{D}}(C)$.

(b) *For all $L \in \mathcal{T}$, $C \subseteq L$ implies $B \subseteq L$.*

Proof. Suppose $\mathcal{C}_{\mathcal{D}}(B) \subseteq \mathcal{C}_{\mathcal{D}}(C)$. Then by Lemma 2.6(c), $C \subseteq L \Leftrightarrow \mathcal{C}_{\mathcal{D}}(C) \subseteq L \Rightarrow \mathcal{C}_{\mathcal{D}}(B) \subseteq L \Leftrightarrow B \subseteq L$ for all $L \in \mathcal{T}$, because $\mathcal{T} \subseteq \mathcal{L}_{\mathcal{D}}$.

On the other hand, assume $C \subseteq L$ implies $B \subseteq L$ for all $L \in \mathcal{T}$. Then $\{L \in \mathcal{T} \mid C \subseteq L\} \subseteq \{L \in \mathcal{T} \mid B \subseteq L\}$, which implies $\mathcal{C}_{\mathcal{D}}(B) \subseteq \mathcal{C}_{\mathcal{D}}(C)$ by Lemmas 2.3 and Proposition 6.24. □

Next we present two simple corollaries of the previous lemma. They are based on the following obvious condition which hold for all $B, C, L \subseteq A$,

$$(C \subseteq L \text{ implies } B \subseteq L) \Leftrightarrow (B - L \neq \emptyset \text{ implies } C - L \neq \emptyset),$$

and the fact that $\mathcal{C}_{\mathcal{D}}$ is a closure operator.

Corollary 6.26 *Let $\mathcal{D} = (A, K)$ be a dependence space. If a subset $\mathcal{T} (\subseteq \wp(\mathcal{A}))$ is dense, then the following conditions are equivalent for all $B, C \subseteq A$.*

(a) $\mathcal{C}_{\mathcal{D}}(B) \subseteq \mathcal{C}_{\mathcal{D}}(C)$.

(b) *For all $L \in \mathcal{T}$, $B - L \neq \emptyset$ implies $C - L \neq \emptyset$.*

Corollary 6.27 *Let $\mathcal{D} = (A, K)$ be a dependence space. If a subset $\mathcal{T}(\subseteq \wp(\mathcal{A}))$ is dense, then the following conditions are equivalent for all $C \subseteq B(\subseteq A)$.*
(a) $\mathcal{C}_{\mathcal{D}}(B) = \mathcal{C}_{\mathcal{D}}(C)$.
(b) *For all $L \in \mathcal{T}$, $B - L \neq \emptyset$ implies $C - L \neq \emptyset$.*

The following proposition characterizes the set $IND_{\mathcal{D}}$ and for all $B \subseteq A$ the set $CORE_{\mathcal{D}}(B)$ by means of dense sets.

Proposition 6.28 *Let $\mathcal{D} = (A, K)$ be a dependence space. If a subset $\mathcal{T}(\subseteq \wp(\mathcal{A}))$ is dense, then the following equations hold for all $B \subseteq A$.*
(a) $IND_{\mathcal{D}} = \{B \in \wp(A) \mid$ *for all* $a \in B,\ B - L = \{a\}$ *for some* $L \in \mathcal{T}\}$.
(b) $CORE_{\mathcal{D}}(B) = \{a \in B \mid B - L = \{a\}$ *for some* $L \in \mathcal{T}\}$.

Proof. Assume $B \subseteq A$ and $a \in B$. Because $B - \{a\} \subseteq B$, we get by Corollary 6.27 that $(B, B - \{a\}) \in K \Leftrightarrow \mathcal{C}_{\mathcal{D}}(B) \neq \mathcal{C}_{\mathcal{D}}(B - \{a\}) \Leftrightarrow$ there exists $L \in \mathcal{T}$ such that $B - L \neq \emptyset$ and $(B - \{a\}) - L = \emptyset \Leftrightarrow B - L = \{a\}$ for some $L \in \mathcal{T}$. This equivalence implies both (a) and (b). □

Out next proposition characterizes the reducts a of given set by applying dense sets.

Proposition 6.29 *Suppose $\mathcal{D} = (A, K)$ is a dependence space and let $\mathcal{T} \subseteq \wp(A)$ be a dense subset. If $B \subseteq A$, then $C \in RED_{\mathcal{D}}(B)$ if and only if C is minimal with respect to the inclusion relation among the subsets of A such that $C \cap (B - L) \neq \emptyset$ for all $L \in \mathcal{T}$ which satisy $B - L \neq \emptyset$.*

Proof. Suppose that $C \in RED_{\mathcal{D}}(B)$. Then $C \subseteq B$ and $\mathcal{C}_{\mathcal{D}}(B) = \mathcal{C}_{\mathcal{D}}(C)$. By Corollary 6.27, $C \cap (B - L) = (B \cap C) - L = C - L \neq \emptyset$ whenever $B - L \neq \emptyset$. If C is not minimal, there is a $C_1 \subset C$ such that $C_1 \cap (B - L) \neq \emptyset$ whenever $(B - L) \neq \emptyset$ for all $L \in \mathcal{T}$. But $C_1 \subset C \subseteq B$ implies $C_1 \cap (B - L) = (C_1 \cap B) - L = C_1 - L$. So, $\mathcal{C}_{\mathcal{D}}(C_1) = \mathcal{C}_{\mathcal{D}}(B) = \mathcal{C}_{\mathcal{D}}(C)$, that is, C is not independent, a contradiction!

Conversely, let C be a minimal subset of A which satisfies $C \cap (B - L) \neq \emptyset$ for all $L \in \mathcal{T}$ such that $B - L \neq \emptyset$. If $C \not\subseteq B$, then $C_1 = (B \cup C) \subset C$ and $C_1 \cap (B - L) = (B \cap C) \cap (B - L) = C \cap (B - L) \neq \emptyset$ whenever $B - L \neq \emptyset$, a contradiction! So, $C \subseteq B$. Since $C - L = C \cap (B - L)$, we get $(B, C) \in K$. Assume $C \notin IND_{\mathcal{D}}$. Then there is a $C_1 \subset C$ such that $\mathcal{C}_{\mathcal{D}}(C_1) \subseteq \mathcal{C}_{\mathcal{D}}(C) = \mathcal{C}_{\mathcal{D}}(B)$. Because $C_1 \subseteq B$, this implies $C_1 \cap (B - L) = C_1 - L \neq \emptyset$ whenever $B - L \neq \emptyset$, a contradiction! □

Note that Proposition 6.29 characterizes the reducts of B as subsets of A (cf. Proposition 5.7). As in Section 5 we shall need this particular characterization later when we are writing an algorithm which computes $RED_{\mathcal{S}}(B)$ for an arbitrary set $B(\subseteq A)$ of attributes.

Example 6.30 Let us consider the dependence space $\mathcal{D}$ of Example 6.20. Now

$$\mathcal{M}(\mathcal{L}_{\mathcal{D}}) = \{\{1, 2, 4\}, \{1, 3\}, \{2, 3\}\}$$

is the least dense set, and $A - L \neq \emptyset$ for all $L \in \mathcal{M}(\mathcal{L}_{\mathcal{D}})\}$.

The reducts of A are the minimal subsets C of A, which satisfy $C \cap (A - L) = C \cap L' = C - L \neq \emptyset$ for every $L \in \mathcal{M}(\mathcal{L}_{\mathcal{D}})$ such that $A - L \neq \emptyset$. Clearly, $RED_{\mathcal{D}}(A) = \{\{3, 4\}, \{1, 2, 3\}\}$ and $CORE_{\mathcal{D}}(A) = \{3\}$.

6.5 Dependency relations in dependence spaces

Here we study dependency relations of dependence space. Note that Propositions 6.32 and 6.34 can be found in [14]. Assume that $\mathcal{D} = (A, K)$ is a dependence space. A subset $B(\subseteq A)$ is said to be *dependent on* $C(\subseteq A)$ in $\mathcal{D}$, which will be denoted by $C \to B$ $(\mathcal{D})$, if $\mathcal{C}_{\mathcal{D}}(B) \subseteq \mathcal{C}_{\mathcal{D}}(C)$. The relation $\to$ $(\mathcal{D})$ is called the *dependency relation of* $\mathcal{D}$. Usually we write simply $C \to B$ instead of $C \to B$ $(\mathcal{D})$ if there is no danger of confusion.

Because $\mathcal{C}_{\mathcal{D}}$ is a closure operator, then $B \subseteq C$ implies $C \to B$. Let $\mathcal{T}$ be a dense subset of $\wp(A)$. Then by Corollary 6.26 the following condition holds for all $B, C \subseteq A$,

(6.5) $\quad C \to B$ if and only if $B - L \neq \emptyset$ implies $C - L \neq \emptyset$ for all $L \in \mathcal{T}$.

In Section 5.3 we presented a solution to the following problem. Let $\mathcal{S} = (U, A, \{V_a\}_{a\in A})$ be an information system in which the sets U, V are finite, and $B, C(\subseteq A)$ satisfy $C \to B$ in $\mathcal{S}$. Then find all minimal subsets D of C which satisfy $D \to B$ $(\mathcal{S})$. Here we give a solution to the corresponding problem in the case of dependence spaces. In [14] is also presented a solution, but our approach essentially differs from it.

Proposition 6.31 *Let $\mathcal{D} = (A, K)$ be a dependence space and assume $\mathcal{T}$ is a dense subset of $\mathcal{D}$. If $C \to B$ holds, then the following conditions are equivalent.*

(a) *D is a minimal subset of C such that $D \to B$.*

(b) *D is a minimal subset of A such that for all $L \in \mathcal{T}$, $D \cap (C - L) \neq \emptyset$ whenever $B - L \neq \emptyset$.*

Proof. Suppose $C \to B$ holds and assume D is a minimal subset of C such that $D \to B$. Because $D \subseteq C$, we get $D = C \cap D$. The assumption $D \to B$ implies $D \cap (C - L) = (C \cap D) - L = D - L \neq \emptyset$ for all $L \in \mathcal{T}$ such that $B - L \neq \emptyset$. Assume D is not minimal, that is, there exists $D_1 \subset D$ which satisfies $D_1 \cap (C - L) \neq \emptyset$ for all $L \in \mathcal{T}$ such that $B - L \neq \emptyset$. But $D_1 \subseteq C$ implies $D_1 - L \neq \emptyset$ for all $L \in \mathcal{T}$ which satisfy $B - L \neq \emptyset$, a contradiction!

Conversely, assume $C \to B$ and that D is a minimal subset of A which satisfies $D \cap (C - L) \neq \emptyset$ whenever $B - L \neq \emptyset$. By this assumption it is clear that $D \subseteq C$. Hence, $D - L = (C \cap D) - L = D \cap (C - L) \neq \emptyset$ for all $L \in \mathcal{T}$ such that $B - L \neq \emptyset$, that is, $D \to B$. Suppose D is not minimal, that is, there exists $D_1 \subset D$ such that $D_1 \to B$. The condition $D_1 \subset D \subseteq C$ implies $D_1 - L = D_1 \cap (C - L) \neq \emptyset$ whenever $B - L \neq \emptyset$, a contradiction! □

Reducts of a subset of a dependence space can be described by means of dependency relation as follows.

Proposition 6.32 *If $\mathcal{D} = (A, K)$ is a dependence space and $B, C \subseteq A$, then C is a reduct of B if and only if C is a minimal subset of B with respect to inclusion relation which satisfies $C \to B$.*

Proof. If C is a reduct of B, then trivially $C \to B$. Suppose there exists a set $C_1 \subset C$ which satisfies $C_1 \to B$, i.e., $\mathcal{C}_{\mathcal{D}}(B) \subseteq \mathcal{C}_{\mathcal{D}}(C_1)$. Because $\mathcal{C}_{\mathcal{D}}$ is a closure operation, then $\mathcal{C}_\mathcal{D}(C_1) \subseteq \mathcal{C}_\mathcal{D}(C) = \mathcal{C}_\mathcal{D}(B)$. Hence, $\mathcal{C}_\mathcal{D}(B) = \mathcal{C}_\mathcal{D}(C_1)$ which implies $(B, C_1) \in K$, a contradiction!

Conversely, assume C is a minimal subset of B which satisfies $C \to B$. Because $\mathcal{C}_\mathcal{D}(B) \subseteq \mathcal{C}_\mathcal{D}(C)$, $C \subseteq B$, and $\mathcal{C}_\mathcal{D}$ is a closure operator, $\mathcal{C}_\mathcal{D}(B) = \mathcal{C}_\mathcal{D}(C)$. Assume C is not independent, that is, there exists a subset $C_1 \subset C(\subseteq B)$ such that $(C_1, C) \in K$. This implies $\mathcal{C}_\mathcal{D}(B) = \mathcal{C}_\mathcal{D}(C_1)$, i.e., $C_1 \to B$, a contradiction! □

If we denote by $\leftarrow$ the inverse of the relation $\to$, then the following result holds.

Lemma 6.33 *If $\mathcal{D} = (A, K)$ is a dependence space, then $K = \to \cap \leftarrow$.*

Proof. For all $B, C \subseteq A$, $(B, C) \in K \Leftrightarrow \mathcal{C}_\mathcal{D}(B) = \mathcal{C}_\mathcal{D}(C) \Leftrightarrow \mathcal{C}_\mathcal{D}(B) \subseteq \mathcal{C}_\mathcal{D}(C)$ and $\mathcal{C}_\mathcal{D}(C) \subseteq \mathcal{C}_\mathcal{D}(B) \Leftrightarrow B \to C$ and $B \leftarrow C$. □

The following proposition characterizes the dependency relations.

Proposition 6.34 *Let A be a finite nonempty set and let r be a binary relation on $\wp(A)$. Then there exists a dependence space $\mathcal{D} = (A, K)$ such that r is its dependency relation if and only if r satisfies the following conditions for all $B, C, D \subseteq A$,*

(a) $B \subseteq C$ *implies* $(C, B) \in r$

(b) $(B, C) \in r$ *and* $(C, D) \in r$ *imply* $(B, D) \in r$.

(c) $(B, C) \in r$ *and* $(B, D) \in r$ *imply* $(B, C \cup D) \in r$.

Proof. Suppose r is a binary relation on $\wp(A)$ which satisfies the conditions (a)–(c). Let us set $K = r \cap r^{-1}$. By (a), $(B, B) \in r$ and $(B, B) \in r^{-1}$ for all $B \subseteq A$. Hence, $(B, B) \in K$, i.e., K is reflexive. If $(B, C) \in K$, then $(B, C) \in r$ and $(B, C) \in r^{-1}$ hold, which implies $(C, B) \in r^{-1}$ and $(C, B) \in r$, i.e., $(C, B) \in K$. Thus, K is symmetric. If $(B, C) \in K$ and $(C, D) \in K$, then $(B, C) \in r$ and $(C, D) \in r$ which implies $(B, D) \in r$ by (b). Similarly, $(B, C) \in r^{-1}$ and $(C, D) \in r^{-1}$ imply $(C, B) \in r$ and $(D, C) \in r$. By (b), $(D, B) \in r$, i.e., $(B, D) \in r^{-1}$. Hence, $(B, D) \in K$, that is, K is an equivalence.

Suppose $(B_1, C_1) \in K$ and $(B_2, C_2) \in K$. The fact $B_1, B_2 \subseteq B_1 \cup B_2$ implies $(B_1 \cup B_2, B_1) \in r$ and $(B_1 \cup B_2, B_2) \in r$ by (a). Then by (b) we get $(B_1 \cup B_2, C_1) \in r$ and $(B_1 \cup B_2, C_2) \in r$. This implies by (c), $(B_1 \cup B_2, C_1 \cup C_2) \in r$. Similarly, $(B_1, C_1) \in K$ and $(B_2, C_2) \in K$ imply $(C_1, B_1) \in r$ and $(C_2, B_2) \in r$. From $C_1, C_2 \subseteq C_1 \cup C_2$ we get $(C_1 \cup C_2, C_1) \in r$ and $(C_1 \cup C_2, C_2) \in r$. Hence, $(C_1 \cup C_2, B_1) \in r$ and $(C_1 \cup C_2, B_2) \in r$, which implies $(C_1 \cup C_2, B_1 \cup B_2) \in r$, that is, $(B_1 \cup B_2, C_1 \cup C_2) \in r^{-1}$. Now $(B_1 \cup B_2, C_1 \cup C_2) \in K$ holds, i.e., K is a congruence on $(\wp(A), \cup)$.

Next we show that for all $B, C \subseteq A$, $(B, C) \in r$ if and only if $B \to C$ $(\mathcal{D})$, where $\mathcal{D} = (A, K)$ and $K = r \cap r^{-1}$. If $B \to C$, then $\mathcal{C}_\mathcal{D}(C) \subseteq \mathcal{C}_\mathcal{D}(B)$. By (a), this implies $(\mathcal{C}_\mathcal{D}(B), \mathcal{C}_\mathcal{D}(C)) \in r$. Because $(B, \mathcal{C}_\mathcal{D}(B)) \in K \subseteq r$ and $(\mathcal{C}_\mathcal{D}(C), C) \in K \subseteq r$, we get $(B, C) \in r$ by (b). On the other hand, suppose $(B, C) \in r$. Since $(\mathcal{C}_\mathcal{D}(B), B) \in K \subseteq r$ and $(C, \mathcal{C}_\mathcal{D}(C)) \in K \subseteq r$, then $(\mathcal{C}_\mathcal{D}(B), \mathcal{C}_\mathcal{D}(C)) \in r$. This implies by (c) that $(\mathcal{C}_\mathcal{D}(B), \mathcal{C}_\mathcal{D}(B) \cup \mathcal{C}_\mathcal{D}(C)) \in r$. Similarly, $\mathcal{C}_\mathcal{D}(B) \subseteq \mathcal{C}_\mathcal{D}(B) \cup \mathcal{C}_\mathcal{D}(C)$ implies $(\mathcal{C}_\mathcal{D}(B) \cup \mathcal{C}_\mathcal{D}(C), \mathcal{C}_\mathcal{D}(B)) \in r$, that is, $(\mathcal{C}_\mathcal{D}(B), \mathcal{C}_\mathcal{D}(B) \cup \mathcal{C}_\mathcal{D}(C)) \in r^{-1}$. Thus, $(\mathcal{C}_\mathcal{D}(B), \mathcal{C}_\mathcal{D}(B) \cup \mathcal{C}_\mathcal{D}(C)) \in K$. Because $\mathcal{C}_\mathcal{D}(B)$ is the greatest element in the congruence class $\mathcal{C}_\mathcal{D}(B)/K$, $\mathcal{C}_\mathcal{D}(B) \subseteq \mathcal{C}_\mathcal{D}(B) \cup \mathcal{C}_\mathcal{D}(C) \subseteq \mathcal{C}_\mathcal{D}(B)$. Hence, $\mathcal{C}_\mathcal{D}(C) \subseteq \mathcal{C}_\mathcal{D}(D)$, that is, $B \to C$.

Conversely, we show that the relation $\to$ satisfies conditions (a)–(c). By the fact that $\mathcal{C}_\mathcal{D}$ is a closure operator, the condition $B \subseteq C$ implies $\mathcal{C}_\mathcal{D}(B) \subseteq \mathcal{C}_\mathcal{D}(C)$, i.e., $C \to B$. If $B \to C$ and $C \to D$, then $\mathcal{C}_\mathcal{D}(D) \subseteq \mathcal{C}_\mathcal{D}(C) \subseteq \mathcal{C}_\mathcal{D}(B)$, which implies $B \to D$. If $B \to C$ and $B \to D$, then $\mathcal{C}_\mathcal{D}(C) \subseteq \mathcal{C}_\mathcal{D}(B)$ and $\mathcal{C}_\mathcal{D}(D) \subseteq \mathcal{C}_\mathcal{D}(B)$, which implies $\mathcal{C}_\mathcal{D}(C \cup D) = \mathcal{C}_\mathcal{D}(\mathcal{C}_\mathcal{D}(C) \cup \mathcal{C}_\mathcal{D}(D)) \subseteq \mathcal{C}_\mathcal{D}(\mathcal{C}_\mathcal{D}(B)) = \mathcal{C}_\mathcal{D}(B)$. Hence, $B \to C \cup D$. □

6.6 Dependence spaces and information systems

In this section we shall see how every information system defines a dependence space, and that for each dependence space there exists an information system which corresponds to this dependence space. We also give detailed methods for these constructions. Let us recall from Section 4.2 the definition of the binary relation $K_\mathcal{S}$ for an information system $\mathcal{S} = (U, A, \{V_a\}_{a \in A})$:

$$K_\mathcal{S} = \{(B, C) \in \wp(A)^2 \mid Ind(B) = Ind(C)\}.$$

By Proposition 4.3 it is obvious that if $\mathcal{S}$ is an information system, then the pair $\mathcal{D}_\mathcal{S} = (A, K_\mathcal{S})$ is a dependence space.

Lemma 6.35 *If $\mathcal{S}$ is an information system, then the following assertions hold.*
(a) $\mathcal{C}_{(\mathcal{D}_\mathcal{S})} = \mathcal{C}_\mathcal{S}$.
(b) $\mathcal{L}_{(\mathcal{D}_\mathcal{S})} = \mathcal{L}_\mathcal{S}$.
(c) $\mathcal{M}(\mathcal{L}_{(\mathcal{D}_\mathcal{S})}) = \mathcal{M}(\mathcal{L}_\mathcal{S})$.

Proof. (a) Assume $B, C \subseteq A$. By Proposition 4.6(c), $\mathcal{C}_\mathcal{S}(B) = \mathcal{C}_\mathcal{S}(\mathcal{C})$ if and only if $(B, C) \in K_\mathcal{S}$, which is equivalent to $\mathcal{C}_{(\mathcal{D}_\mathcal{S})}(B) = \mathcal{C}_{(\mathcal{D}_\mathcal{S})}(C)$ by (6.3). Statements (b) and (c) follow easily from (a) by the fact that there exists a bijective relationship between closure operators and closure systems. □

We have seen that in a dependence space $\mathcal{D}$ the function $\mathcal{C}_\mathcal{D}$ can be computed from any dense subset $\mathcal{T}(\subseteq \wp(A))$. In the following we see how every information system $\mathcal{S} = (U, A, \{V_a\}_{a \in A})$ determines a dense subset of $\wp(A)$ in the dependence space $\mathcal{D}_\mathcal{S} = (A, K_\mathcal{S})$. First we shall present this useful lemma. Here $-\!\!\!\subset$ denotes the covering relation of $(\mathcal{L}_\mathcal{S}, \subseteq)$.

Lemma 6.36 *Let $\mathcal{S} = (U, A, \{V_a\}_{a \in A})$ be an information system in which $U = \{x_1, \ldots, x_n\}$ and $A = \{a_1, \ldots, a_m\}$. If $(c_{ij})_{n \times n}$ is the discernibility matrix of $\mathcal{S}$ and $1 \leq i < j \leq n$, then the following holds.*
(a) $(x_i, x_j) \in Ind(A - c_{ij})$.,
(b) *$A - c_{ij}$ is the greatest subset B of A which satisfies $(x_i, x_j) \in Ind(B)$.*
(c) *If $C \in \mathcal{M}(\mathcal{L}_\mathcal{S})$, then there exists $c \in A - C$ such that $Ind(C \cup \{a\}) \subseteq Ind(C \cup \{c\}) \subset Ind(C)$ for all $a \in A - C$.*

Proof. (a) Because $a(x_i) = a(x_j)$ for all $a \in A - c_{ij}$, $(x_i, x_j) \in \ker a$ for all $a \in A - c_{ij}$. Hence, $(x_i, x_j) \in \bigcap_{a \in (A - c_{ij})} \ker a = Ind(A - c_{ij})$.

(b) Let B be a subset of A such that $(x_i, x_j) \in Ind(B)$. Then for all $a \in B$, $(x_i, x_j) \in Ind(B) \subseteq Ind(\{a\})$. This implies $a(x_i) = a(x_j)$, i.e., $a \in A - c_{ij}$.

(c) Suppose $C \in \mathcal{M}(\mathcal{L}_\mathcal{S})$. Because $C \in \mathcal{L}_\mathcal{S}$, then for all $a \in A - C$ holds $Ind(C \cup \{a\}) \subset Ind(C)$. Because $\mathcal{L}_\mathcal{S}$ is finite, then there exists exactly one $D \in \mathcal{L}_\mathcal{S}$ such that $C -\!\!\subset D$ in $\mathcal{L}_\mathcal{S}$. Suppose $c \in D - C$. Then $Ind(D) = Ind(C \cup \{c\}) \subset Ind(C)$ holds.

Assume $a \in A - C$ and let us denote $C_a = \mathcal{C}_\mathcal{S}(C \cup \{a\})$. This implies $Ind(C_a) = Ind(C \cup \{a\}) \subset Ind(C)$. If $D \not\subseteq C_a$, then $C \subseteq D \cap C_a \subset D$. Because $C -\!\!\subset D$ in $\mathcal{L}_\mathcal{S}$, then either (i) $C = D$ or (ii) $C = C_a$ holds. But (i) implies $Ind(C) = Ind(C \cup \{c\})$ a contradiction! Similarly, (ii) implies $Ind(C) = Ind(C \cup \{a\})$, a contradiction! Hence, $D \subseteq C_a$ which implies $Ind(C \cup \{a\}) = Ind(C_a) \subseteq Ind(D) = Ind(C \cup \{c\})$ for all $a \in A - C$. □

Proposition 6.37 *Let $\mathcal{S} = (U, A, \{V_a\}_{a \in A})$ be an information system in which $U = \{x_1, \ldots, x_n\}$, $A = \{a_1, \ldots, a_m\}$, and let $(c_{ij})_{n \times n}$ be the discernibility matrix of $\mathcal{S}$. If $(\mathcal{L}_\mathcal{S}, \subseteq)$ is the closure lattice of $\mathcal{S}$, then the following holds.*

(a) *For all $1 \leq i < j \leq n$, $(A - c_{ij}) \in \mathcal{L}_\mathcal{S}$.*

(b) *If $C \in \mathcal{M}(\mathcal{L}_\mathcal{S})$, then $C = A - c_{ij}$ for some $1 \leq i < j \leq n$.*

Proof. Now $\mathcal{C}_\mathcal{S}(A - c_{ij}) = \{a \in A \mid Ind(A - c_{ij}) \subseteq Ind(\{a\})\}$. It is clear that $(A - c_{ij}) \subseteq \mathcal{C}_\mathcal{S}(A - c_{ij})$ for all $1 \leq i < j \leq n$. We have to show that $\mathcal{C}_\mathcal{S}(A - c_{ij}) \subseteq (A - c_{ij})$ holds. Suppose $a \in \mathcal{C}_\mathcal{S}(A - c_{ij})$. Then $Ind(A - c_{ij}) \subseteq Ind(\{a\})$. By Lemma 6.36(a), $(x_i, x_j) \in Ind(A - c_{ij})$, which implies $(x_i, x_j) \in Ind(\{a\})$, i.e., $a(x_i) = a(x_j)$. Thus, $a \in A - c_{ij}$.

(b) Assume $C \in \mathcal{M}(\mathcal{L}_\mathcal{S})$, Then by Lemma 6.36(c), there exists $c \in A - C$ such that $Ind(C \cup \{a\}) \subseteq Ind(C \cup \{c\}) \subset Ind(C)$ for all $a \in A - C$. Because $Ind(C \cup \{c\}) \subset Ind(C)$, there exists (x_i, x_j), $1 \leq i < j \leq n$ such that $(x_i, x_j) \in Ind(C)$ and $(x_i, x_j) \notin Ind(C \cup \{c\})$. Because $a(x_i) = a(x_j)$ for all $a \in C$, $C \subseteq A - c_{ij}$. Next we show that $(A - c_{ij}) \subseteq C$. If $a \in A - c_{ij}$, then $(x_i, x_j) \in Ind(\{a\})$. Suppose $a \notin C$. We know that $(x_i, x_j) \in Ind(C)$. Hence, $(x_i, x_j) \in Ind(C) \cap Ind(\{a\}) = Ind(C \cup \{a\})$. Because for all $a \in A - C$ holds $Ind(C \cup \{a\}) \subseteq Ind(C \cup \{c\})$, we get $(x_i, x_j) \in Ind(C \cup \{c\})$, a contradiction! Hence, $(A - c_{ij}) \subseteq C$, which implies $C = (A - c_{ij})$. □

Proposition 6.37 has the following corollary.

Corollary 6.38 *Suppose $\mathcal{S} = (U, A, \{V_a\}_{a \in A})$ is an information system in which $U = \{x_1, \ldots, x_n\}$, $A = \{a_1, \ldots, a_m\}$, and $(c_{ij})_{n \times n}$ is the discernibility matrix of $\mathcal{S}$. Then the set $\{A - c_{ij} \mid 1 \leq i < j \leq n\}$ is dense in the dependence space $\mathcal{D}_\mathcal{S} = (A, K_\mathcal{S})$.*

Example 6.39 The discernibility matrix of the information system $\mathcal{S}$ in Example 4.1 is presented in Example 5.5. It can be easily computed that

$$\{A - c_{ij} \mid 1 \leq i < j \leq n\} = \{\{1, 3\}, \{3\}, \{1\}, \{2, 3\}, \{1, 2, 4\}, \{2\}\}.$$

Moreover,

$$\mathcal{M}(\mathcal{L}_\mathcal{S}) \subseteq \{A - c_{ij} \mid 1 \leq i < j \leq n\} \subseteq \mathcal{L}_\mathcal{S}.$$

We have shown how a dense subset of the dependence space $\mathcal{D}_\mathcal{S} = (A, K_\mathcal{S})$ can be determined from the discernibility matrix of a given finite information system $\mathcal{S}$.

Next we shall show that for every dependence space $\mathcal{D}$ there is an information system $\mathcal{S}_{\mathcal{D}}$ such that $\mathcal{D} = \mathcal{D}_{\mathcal{S}_{\mathcal{D}}}$. Our construction is modified from the one presented in [14] which is not so precise.

Let $\mathcal{D} = (A, K)$ be a dependence space such that $\mathcal{M}(\mathcal{L}_{\mathcal{D}}) = \{L_1, \ldots, L_k\}$ and let us put $U = \{x_1, \ldots, x_{k+1}\}$. We define a function $h : U^2 \to \wp(A)$ as follows:

1. $h(x_i, x_i) = A$ for $1 \leq i \leq k+1$.
2. $h(x_1, x_i) = L_{i-1}$ for $2 \leq i \leq k+1$.
3. $h(x_i, x_j) = h(x_1, x_i) \cap h(x_1, x_j)$ for $2 \leq i < j \leq k+1$.
4. $h(x_j, x_i) = h(x_i, x_j)$ for $2 \leq i < j \leq k+1$.

Further, for all $a \in A$, we define a binary relation r_a on U by

$$(x_i, x_j) \in r_a \text{ if and only if } a \in h(x_i, x_j),$$

for all $1 \leq i, j \leq k+1$. Then the following lemma holds.

Lemma 6.40 *Let $\mathcal{D} = (A, K)$ be a dependence space such that $\mathcal{M}(\mathcal{L}_{\mathcal{D}}) = \{L_1, \ldots, L_k\}$ and let us put $U = \{x_1, \ldots, x_{k+1}\}$. Then for all $a \in A$, the relation r_a is an equivalence on U.*

Proof. Because for all $1 \leq i \leq k+1$ and $a \in A$, $a \in h(x_i, x_i)$ which implies $(x_i, x_i) \in r_a$. Hence, r_a is reflexive. Suppose $(x_i, x_j) \in r_a$. Then $a \in h(x_i, x_j) = h(x_j, x_i)$. Thus, $(x_j, x_i) \in r_a$, i.e., r_a is symmetric. Suppose $(x_i, x_j) \in r_a$ and $(x_j, x_l) \in r_a$. Without any loss of generality we may assume $i \leq j \leq l$. Because $a \in h(x_i, x_j)$ and $a \in h(x_j, x_l)$ imply $a \in h(x_1, x_i) \cap h(x_1, x_j)$ and $a \in h(x_1, x_j) \cap h(x_1, x_l)$, we have $a \in h(x_1, x_i) \cap h(x_1, x_l)$. This implies $a \in h(x_i, x_l)$, that is, r_a is transitive. □

Example 6.41 Let us consider the dependence space $\mathcal{D}$ of Example 6.20 in which

$$\mathcal{M}(\mathcal{L}_{\mathcal{D}}) = \{\{1,2,4\}, \{1,3\}, \{2,3\}\}.$$

If we set $U = \{x_1, x_2, x_3, x_4\}$, then we may define a function $h : U^2 \to \mathcal{M}(\mathcal{L}_{\mathcal{D}})$, which is presented in Table 3.

	x_1	x_2	x_3	x_4
x_1	A	$\{1,2,4\}$	$\{1,3\}$	$\{2,3\}$
x_2	$\{1,2,4\}$	A	$\{1\}$	$\{2\}$
x_3	$\{1,3\}$	$\{1\}$	A	$\{3\}$
x_4	$\{2,3\}$	$\{2\}$	$\{3\}$	A

Table 3

The equivalence classes of equivalences r_a, where $a \in A$, are:

- $U/r_1 = \{\{1,2,3\}, \{4\}\} =: \{b_1^1, b_2^1\}$,
- $U/r_2 = \{\{1,2,4\}, \{3\}\} =: \{b_1^2, b_2^2\}$,

- $U/r_3 = \{\{1,3,4\},\{2\}\} =: \{b_1^3, b_2^3\}$,
- $U/r_4 = \{\{1,2\},\{3\},\{4\}\} =: \{b_1^4, b_2^4, b_3^4\}$.

As before, let us denote by v_e the canonical map $U \to U/e$, $x \mapsto x/e$, of an equivalence e.

Proposition 6.42 *Let $\mathcal{D} = (A, K)$ be a dependence space in which $\mathcal{M}(\mathcal{L}_{\mathcal{D}}) = \{L_1, \ldots, L_k\}$ and let us put $U = \{x_1, \ldots, x_{k+1}\}$. If we set $a^* = v_{r_a}$, $V_{a^*} = U/r_a$ for all $a \in A$, and $B^* = \{a^* \mid a \in B\}$ for all $B \subseteq A$, then $\mathcal{S}_{\mathcal{D}} = (U, A^*, \{V_a\}_{a \in A^*})$ is an information system such that*

$$(B, C) \in K \text{ if and only if } (B^*, C^*) \in K_{(\mathcal{S}_{\mathcal{D}})}.$$

Proof. Suppose $(B, C) \in K$ holds. Then $\mathcal{C}_{\mathcal{D}}(B) = \mathcal{C}_{\mathcal{D}}(C)$, and for all $L \in \mathcal{M}(\mathcal{C}_{\mathcal{D}})$ the conditions $B \subseteq L$ and $C \subseteq L$ are equivalent. It is obvious that $\ker a^* = \ker v_{r_a} = r_a$. Hence, for all $B \subseteq A$, $Ind(B^*) = \bigcap_{a^* \in B^*} \ker a^* = \bigcap_{a \in B} r_a$. Suppose $(x_i, x_j) \in Ind(B^*)$. Then $(x_i, x_j) \in \bigcap_{a \in B} r_a$, that is, $(x_i, x_j) \in r_a$ for all $a \in B$. Hence, $B \subseteq h(x_i, x_j) = h(x_1, x_i) \cap h(x_1, x_j)$. Because $h(x_1, x_i) \in \mathcal{M}(\mathcal{C}_{\mathcal{D}})$ and $h(x_1, x_j) \in \mathcal{M}(\mathcal{C}_{\mathcal{D}})$, this implies $C \subseteq h(x_i, x_j) = h(x_1, x_i) \cap h(x_1, x_j)$. Then $(x_i, x_j) \in \bigcap_{a \in C} r_a = Ind(C^*)$. Similarly, we can show that $Ind(C^*) \subseteq Ind(B^*)$. Hence, $Ind(B^*) = Ind(C^*)$, which implies $(B^*, C^*) \in K_{(\mathcal{S}_{\mathcal{D}})}$.

Conversely, if $(B^*, C^*) \in K_{(\mathcal{S}_{\mathcal{D}})}$, then $Ind(B^*) = Ind(C^*)$. Hence, for all (x_1, x_i), $2 \le i \le k+1$, $(x_1, x_i) \in Ind(B^*) = \bigcap_{a \in B} r_a$ if and only if $(x_1, x_i) \in Ind(C^*) = \bigcap_{a \in C} r_a$, that is, $B \subseteq h(x_1, h_i)$ if and only if $C \subseteq h(x_1, h_i)$ for all $2 \le i \le k+1$. Because $\mathcal{M}(\mathcal{L}_{\mathcal{D}}) = \{h(x_1, x_i) \mid 2 \le i \le k+1\}$, we get $B \subseteq L$ if and only if $C \subseteq L$ for all $L \in \mathcal{M}(\mathcal{C}_{\mathcal{D}})$, which implies $(B, C) \in K$. ☐

Example 6.43 The information system $\mathcal{S}_{\mathcal{D}}$ corresponding to the dependence space $\mathcal{D}$ of Example 6.20 can be represented by Table 4. The values b_i^j refer to Example 6.41.

	1^*	2^*	3^*	4^*
x_1	b_1^1	b_1^2	b_1^3	b_1^4
x_2	b_1^1	b_1^2	b_2^3	b_1^4
x_3	b_1^1	b_2^2	b_1^3	b_2^4
x_4	b_2^1	b_1^2	b_1^3	b_3^4

Table 4

Chapter 7

A representation for dependence spaces

7.1 Difference functions in dependence spaces

In this section we study difference functions which help us to write an algorithm for the reduction problem that is in many cases more efficient than those presented in [12, 13]. The notion of difference functions is introduced in [8]. However, here we give an equivalent, but a clearer definition.

Suppose that $\mathcal{D} = (A, K)$ is a dependence space in which $A = \{a_1, \ldots, a_m\}$. For any $B \subseteq A$, let $\delta(B)$ denote the disjunction of all variables y_i, where $a_i \in B$. We define the *difference function* $f_B^{\mathcal{D}}(y_1, \ldots, y_m)$ of B as the conjunction

$$\bigwedge_{\substack{L \in \mathcal{M}(\mathcal{L}_{\mathcal{D}}) \\ B - L \neq \emptyset}} \delta(B - L).$$

It is clear that the function $f_B^{\mathcal{D}}$ is isotone. Since $\bigwedge \emptyset = 1$, $f_B^{\mathcal{D}} = \top \Leftrightarrow B \subseteq L$ for all $L \in \mathcal{M}(\mathcal{L}_{\mathcal{D}}) \Leftrightarrow (B, \emptyset) \in K$.

By the definition of $f_B^{\mathcal{D}}$ we can now write the following conditions for every $B, C (\subseteq A)$.

(7.1) $\quad f_B^{\mathcal{D}}(\chi(C)) = 1 \Leftrightarrow C \cap (B - L) \neq \emptyset$ for all $L \in \mathcal{M}(\mathcal{L}_{\mathcal{D}})$ such that $B - L \neq \emptyset$.

(7.2) $\quad f_B^{\mathcal{D}}(\chi(C)) = 0 \Leftrightarrow C \subseteq (B - L)'$ for some $L \in \mathcal{M}(\mathcal{L}_{\mathcal{D}})$ such that $B - L \neq \emptyset$.

Our next proposition follows easily from Proposition 6.29 and (7.1) and (7.2).

Proposition 7.1 *If $\mathcal{D} = (A, K)$ is a dependence space and $B \subseteq A$, then*
(a) $\min T(f_B^{\mathcal{D}}) = \{\chi(C) \mid C \in RED_{\mathcal{D}}(B)\}$, *and*
(b) $\max F(f_B^{\mathcal{D}}) = \max\{\chi((B - L)') \mid L \in \mathcal{M}(\mathcal{L}_{\mathcal{D}}),\ B - L \neq \emptyset\}$. □

Also the following corollary is obvious.

Corollary 7.2 *If $\mathcal{D} = (A, K)$ is a dependence space and $B \subseteq A$, then $\{a_{i_1}, \ldots, a_{i_p}\}$ is a reduct of B if and only if $y_{i_1} \wedge \cdots \wedge y_{i_p}$ is a prime implicant of $f_B^{\mathcal{D}}$.*

Example 7.3 In Example 6.20,

$$\{A - L \mid A - L \neq \emptyset,\ L \in \mathcal{M}(\mathcal{L}_\mathcal{D})\} = \{\{3\}, \{2,4\}, \{1,4\}\}.$$

Hence,

$$f_A^\mathcal{D} = 3 \wedge (2 \vee 4) \wedge (1 \vee 4) = 3 \wedge (4 \vee (1 \wedge 2)) = (3 \wedge 4) \vee (1 \wedge 2 \wedge 3),$$

where i stands for y_i. The function $f_A^\mathcal{D}$ has obviously the prime implicants $(3 \wedge 4)$ and $(1 \wedge 2 \wedge 3)$, which implies $RED_\mathcal{D}(A) = \{\{3,4\}, \{1,2,3\}\}$.

7.2 Dependency functions in dependence spaces

Here we introduce the notion of dependency functions in the context of dependence space. They enable us to give a method for finding for a dependency $C \to B$ the set of all minimal subsets D of C which satisfy $D \to B$.

Suppose that $\mathcal{D} = (A, K)$ is a dependence space in which $A = \{a_1, \ldots, a_m\}$. If the dependency $C \to B$ holds in $\mathcal{D}$, then we define the *difference function* $f^\mathcal{D}_{C\to B}(y_1, \ldots, y_m)$ of the dependency $C \to B$ as the conjunction

$$\bigwedge_{\substack{L \in \mathcal{M}(\mathcal{L}_\mathcal{D}) \\ B - L \neq \emptyset}} \delta(C - L).$$

Obviously, the function $f^\mathcal{D}_{C\to B}$ is isotone, and $f^\mathcal{D}_B = \top$ if and only if $(B, \emptyset) \in K$.

We can now write the following conditions for every $B, C (\subseteq A)$.

(7.3) $f^\mathcal{D}_{C\to B}(\chi(D)) = 1 \Leftrightarrow D \cap (C - L) \neq \emptyset$ for all $L \in \mathcal{M}(\mathcal{L}_\mathcal{D})$ such that $B - L \neq \emptyset$.

(7.4) $f^\mathcal{D}_{C\to B}(\chi(D)) = 0 \Leftrightarrow D \subseteq (C - L)'$ for some $L \in \mathcal{M}(\mathcal{L}_\mathcal{D})$ such that $B - L \neq \emptyset$.

The following proposition follows easily from (7.3), (7.4), and Proposition 6.31.

Proposition 7.4 *If $\mathcal{D} = (A, K)$ is a dependence space and $B \subseteq A$, then*
(a) $\min T(f^\mathcal{D}_{C\to B}) = \{\chi(D) \mid D$ *is a minimal subset of C such that* $D \to B\}$, *and*
(b) $\max F(f^\mathcal{D}_{C\to B}) = \max\{\chi((C - L)') \mid L \in \mathcal{M}(\mathcal{L}_\mathcal{D}),\ B - L \neq \emptyset\}$. □

Now we can write the following corollary.

Corollary 7.5 *If $\mathcal{D} = (A, K)$ is a dependence space and the dependency $C \to B$ holds in $\mathcal{D}$, then $D = \{a_{i_1}, \ldots, a_{i_p}\}$ is a minimal subset of C such that $D \to B$ holds if and only if $y_{i_1} \wedge \cdots \wedge y_{i_p}$ is a prime implicant of $f^\mathcal{D}_{C\to B}$.*

Example 7.6 Let us consider the dependence space $\mathcal{D}$ of Example 6.20. If we set $B = \{4\}$, then $A \to B$. Obviously

$$\{A - L \mid B - L \neq \emptyset,\ L \in \mathcal{M}(\mathcal{L}_\mathcal{D})\} = \{\{2,4\}, \{1,4\}\}.$$

The dependency function of the dependency $A \to B$ is

$$f^\mathcal{D}_{A\to B} = (1 \vee 4) \wedge (2 \vee 4) = 4 \vee (1 \wedge 2),$$

where i stands for y_i. The function $f^\mathcal{D}_{A\to B}$ has obviously the prime implicants 4 and $(1 \wedge 2)$, which implies that $\{4\}$ and $\{1,2\}$ are the minimal subsets D of A which satisfy $A \to B$.

7.3 A data type and basic algorithms for dependence spaces

In this section we present a simple implementation of dependence spaces as a data type, which is sufficient for us to solve problems concerning cores, dependence relations, independent sets and reducts.

By Proposition 6.24 we can compute the closure $\mathcal{C}_{\mathcal{D}}(B)$ from any dense set $\mathcal{T}(\subseteq \wp(A))$ of a dependence space $\mathcal{D}$. As we have seen, $\mathcal{M}(\mathcal{L}_{\mathcal{D}})$ is the least dense set. Hence, the simplest way to represent a dependence space $\mathcal{D} = (A, K)$, where $A = \{a_1, \ldots, a_m\}$, is to give a set of vectors $M_{\mathcal{D}}$ which corresponds to the sets in $\mathcal{M}(\mathcal{L}_{\mathcal{D}})$, i.e., $v \in M_{\mathcal{D}}$ if and only if $v = \chi(L)$ for some $L \in \mathcal{M}(\mathcal{L}_{\mathcal{D}})$. The space needed to represent $\mathcal{D}$ in this manner is $O(m|\mathcal{M}(\mathcal{L}_{\mathcal{D}})|)$.

Example 7.7 The dependence space $\mathcal{D}$ in Example 3.9 can be represented as a set of vectors $\{(1,1,0,1), (1,0,1,0), (0,1,1,0)\}$.

The following algorithm which finds the vector corresponding to the set $\mathcal{C}_{\mathcal{D}}(B)$ for any $B(\subseteq A)$ is based on Proposition 6.24. The complexity of this algorithm is $O(m|\mathcal{M}(\mathcal{L}_{\mathcal{D}})|)$.

Algorithm 7.8 CLOSURE
Input: $M = \{\chi(L) \mid L \in \mathcal{M}(\mathcal{L}_{\mathcal{D}})\}$ and a vector $b = \chi(B)$.
Output: $c = \chi(\mathcal{C}_{\mathcal{D}}(B))$.

1. Start with $c := \mathbf{1}_m$.

2. For all $v \in M$, if $b \leq v$, then $c := c \wedge v$.

3. Output c.

By (6.3), $(B, C) \in K$ if and only if $\mathcal{C}_{\mathcal{D}}(B) = \mathcal{C}_{\mathcal{D}}(C)$. Hence, the test whether $(B, C) \in K$ holds, takes $O(m|\mathcal{M}(\mathcal{L}_{\mathcal{D}})|)$ steps, for we may form the closures $\mathcal{C}_{\mathcal{D}}(B), \mathcal{C}_{\mathcal{D}}(C)$ and then check whether they are equal or not. Similarly, the complexity of the test whether the dependency $C \to B$ $(\mathcal{D})$ holds is $O(m|\mathcal{M}(\mathcal{L}_{\mathcal{D}})|)$.

The following algorithm computes the set $CORE_{\mathcal{D}}(B)$ for any $B \subseteq A$. Observe that we could compute the core of B also by the condition $a \in CORE_{\mathcal{D}}(B)$ if and only if $\mathcal{C}_{\mathcal{D}}(B) \neq \mathcal{C}_{\mathcal{D}}(B - \{a\})$, and by applying Algorithm CLOSURE. But this method takes $O(m^2|\mathcal{M}(\mathcal{L}_{\mathcal{D}})|)$ time, while the complexity of the following algorithm is $O(m|\mathcal{M}(\mathcal{L}_{\mathcal{D}})|)$. It is based on Proposition 6.28(b).

Algorithm 7.9 CORE
Input: $M = \{\chi(L) \mid L \in \mathcal{M}(\mathcal{L}_{\mathcal{D}})\}$ and $b = \chi(B)$.
Output: $\chi(CORE_{\mathcal{D}}(B))$.

1. Start with $c = \mathbf{0}_m$.

2. For all $v \in M$, if $b - v$ contains exactly one 1 and this is in the ith position, then $c[i] := 1$.

3. Output c.

By applying Algorithm CORE it is easy to decide whether a subset B is independent or not. Namely, $B \in IND_{\mathcal{D}}$ if and only if $B = CORE_{\mathcal{D}}(B)$. Obviously this test requires $O(m|\mathcal{M}(\mathcal{L}_{\mathcal{D}})|)$ time. Next we give Algorithm REDUCTS which computes on reducts of an arbitrary subset.

Algorithm 7.10 REDUCT
Input: $M = \{\chi(L) \mid L \in \mathcal{M}(\mathcal{L}_{\mathcal{D}})\}$ and $b = \chi(B)$.
Output: $\chi(C)$ for some $C \in RED_{\mathcal{D}}(B)$.

1. Start with $c := b$.
2. For all $i := 1, \ldots, m$, let $c[i] := 0$ if for all $v \in M$, $b - v \neq \mathbf{0}_m$ implies $c[c_i = 0] - v \neq \mathbf{0}_m$.
3. Output c.

The complexity of Algorithm REDUCT is $O(m^2|\mathcal{M}(\mathcal{L}_{\mathcal{D}})|)$

We have seen that the set of all minimal true vector of the difference function $f_B^{\mathcal{D}}$ is the set of characteristic vectors of the reducts of B. Similarly, for a dependency $C \to B$ the set of all minimal true vectors of the function $f_{C \to B}^{\mathcal{D}}$ is the set of characteristic vectors of the minimal subsets D of C such that $D \to B$ holds.

The following algorithm computes the set $\max F(f_B^{\mathcal{D}})$. It is based on Proposition 7.1(b).

Algorithm 7.11 MF-VECTORS3
Input: $M = \{\chi(L) \mid L \in \mathcal{M}(\mathcal{L}_{\mathcal{D}})\}$ and $b = \chi(B)$.
Output: $\max F(f_B^{\mathcal{D}})$.

1. Start with $MF := \emptyset$. For all $v \in M$, if $b - v \neq \mathbf{0}_m$, then $MF := MF \cup \{(b-v)'\}$.
2. Delete from MF all vectors which are not maximal.
3. Output MF.

The complexity of the previous algorithm is $O(m|\mathcal{M}(\mathcal{L}_{\mathcal{D}})|^2)$. Which is the time needed by the dominating Step 2. It is obvious that $|\max F(f_B^{\mathcal{D}}) \leq |\mathcal{M}(\mathcal{L}_{\mathcal{D}})|$. Our next algorithm computes the set $\max F(f_{C \to B}^{\mathcal{D}})$ of the dependency function $f_{C \to B}^{\mathcal{D}}$. The method is based on Proposition 7.4(b).

Algorithm 7.12 MF-VECTORS4
Input: $M = \{\chi(L) \mid L \in \mathcal{M}(\mathcal{L}_{\mathcal{D}})\}$ and vectors $b = \chi(B)$ and $c = \chi(C)$ which satisfy $C \to B$ $(\mathcal{D})$.
Output: $\max F(f_{C \to B}^{\mathcal{D}})$.

1. Start with $MF := \emptyset$. For all $v \in M$, if $b - v \neq \mathbf{0}_m$, then $MF := MF \cup \{(c-v)'\}$.
2. Delete from MF all vectors which are not maximal.
3. Output MF.

The complexity of the previous algorithm is $O(m|\mathcal{M}(\mathcal{L}_\mathcal{D})|^2)$.

Now we can present an algorithm, which finds the reducts of a given subset B of a dependence space $\mathcal{D} = (A, K)$.

Algorithm 7.13 REDUCTS
Input: $M = \{\chi(L) \mid L \in \mathcal{M}(\mathcal{L}_\mathcal{D})\}$ and $b = \chi(B)$.
Output: A set of vectors corresponding to $RED_\mathcal{D}(B)$.

1. Compute the set $\max F(f_B^\mathcal{D})$ with Algorithm MF-VECTORS3.
2. Compute the set $\min T(f_B^\mathcal{D})$ with Algorithm MT-VECTORS and output it.

We already know that $|\max F(f_B^\mathcal{D})| \leq |\mathcal{M}(\mathcal{L}_\mathcal{D})|$ and $|\min T(f_B^\mathcal{D})| = |RED_\mathcal{D}(B)|$. Because the complexity of Step 1 is $O(m|\mathcal{M}(\mathcal{L}_\mathcal{D})|^2)$, the total running time of Algorithm REDUCTS is

$$O(m|RED_\mathcal{D}(B)|(m(|RED_\mathcal{D}(B)|^2+|\mathcal{M}(\mathcal{L}_\mathcal{D})|^2)+T_{EQ}(m(|RED_\mathcal{D}(B)|+|\mathcal{M}(\mathcal{L}_\mathcal{D})|)))).$$

Example 7.14 As we have seen, the dependence space of Example 3.9 can be represented as a set $M = \{(1,1,0,1),(1,0,1,0),(0,1,1,0)\}$ of vectors. We show how Algorithm REDUCTS computes the reducts of the vector $(1,1,1,1)$, which corresponds to the set $\{1,2,3,4\}$. First we compute the set $\max F(f_A^\mathcal{D})$ with Algorithm MF-VECTORS3. Because $\chi(A) - v \neq \mathbf{0}_m$ for all $v \in M$,

$$MF := \{(1,1,0,1), (1,0,1,0), (0,1,1,0)\}.$$

Clearly all vectors in MF are maximal which implies

$$\max F(f_B^\mathcal{D}) := \{(1,1,0,1), (1,0,1,0), (0,1,1,0)\}.$$

Next we compute the set $\min T(f_A^\mathcal{D})$ with Algorithm MT-SETS. This computation is already presented in Example 5.28. Hence, $\min T(f_A^\mathcal{D}) = \{(0,0,1,1), (1,1,1,0)\}$, which implies that $(0,0,1,1)$ and $(1,1,1,0)$ are the vectors corresponding to the reducts of A.

If the dependency $C \to B$ holds in $\mathcal{D}$, then the following algorithm finds the set of characteristic vectors of all minimal subsets D of C which satisfy $D \to B$.

Algorithm 7.15 MIN-DEPENDENCY
Input: $M = \{\chi(L) \mid L \in \mathcal{M}(\mathcal{L}_\mathcal{D})\}$ and two vectors $b = \chi(B)$ and $c = \chi(C)$ such that $C \to B$ in $\mathcal{D}$.
Output: $\{\chi(D) \mid D$ is a minimal subset of C which satisfies $D \to B\}$.

1. Compute the set $\max F(f_{C\to B}^\mathcal{D})$ with Algorithm MF-VECTORS4.
2. Compute the set $\min T(f_{C\to B}^\mathcal{D})$ with Algorithm MT-VECTORS and output it.

Obviously, $|\max F(f_{C\to B}^\mathcal{D})| \leq |\mathcal{M}(\mathcal{L}_\mathcal{D})|$ and if we denote $k = |\{D \mid D$ is a minimal subset of C which satisfies $D \to B\}|$, then the running time of Algorithm MIN-DEPENDENCY is

$$O(mk(m(k^2 + |\mathcal{M}(\mathcal{L}_\mathcal{D})|^2) + T_{EQ}(m(k + |\mathcal{M}(\mathcal{L}_\mathcal{D})|)))).$$

Example 7.16 The dependence space of Example 6.24 can be represented as a set $M = \{(1,1,0,1),(1,0,1,0),(0,1,1,0)\}$ of vectors. Let us set $B = \{4\}$. We show how Algorithm MIN-DEPENDENCY computes the set of the characteristic vectors of all minimal subsets D of A which satisfy $D \to B$. First we compute the set $\max F(f^{\mathcal{D}}_{A\to B})$ with Algorithm MF-VECTORS4. First,

$$MF := \{(\chi(A) - v)' \mid b - v \neq \mathbf{0}_m \text{ for all } v \in M\} = \{(1,0,1,0), (0,1,1,0)\}.$$

Obviously, both vectors in MF are maximal which implies

$$\max F(f^{\mathcal{D}}_{A\to B}) := \{(1,0,1,0), (0,1,1,0)\}.$$

In Example 5.30 it was already presented how we can compute the set $\min T(f^{\mathcal{D}}_{A\to B})$ with Algorithm MT-VECTORS. Obviously, $\min T(f^{\mathcal{D}}_{A\to B}) = \{(0,0,0,1), (1,1,0,0)\}$, which implies that $\{4\}$ and $\{1,2\}$ are the minimal subsets D of A which satisfy $D \to B$.

Suppose $\mathcal{S} = (U, A, \{V_a\}_{a\in A})$ is an information system such that $U = \{x_1,\dots,x_n\}$, $A = \{a_1,\dots,a_m\}$, and $(c_{ij})_{n\times n}$ is the discernibility matrix of $\mathcal{S}$, then by Corollary 6.38, the set $\{A - c_{ij} \mid 1 \leq i < j \leq n\}$ is dense in $\mathcal{D}_{\mathcal{S}} = (A, K_{\mathcal{S}})$. Especially,

$$\mathcal{M}(\mathcal{L}_{(\mathcal{D}_{\mathcal{S}})}) \subseteq \{A - c_{ij} \mid 1 \leq i < j \leq n\}.$$

By this fact it is now easy to write an algorithm which computes from the representation of an information system $\mathcal{S}$ the presentation of the dependence space $\mathcal{D}_{\mathcal{S}}$.

If $V = \{v_1,\dots,v_l\}$ is a set of vectors, then $\bigwedge\{v_1, v_2,\dots,v_l\}$ is an abbreviation for $v_1 \wedge (v_2 \wedge \dots \wedge (v_{l-1} \wedge v_l))$. If $V = \emptyset$, then $\bigwedge V = \mathbf{1}_m$. This operation is used in Step 2 of the algorithm. The complexity of $\bigwedge V$ is $O(m|V|)$.

Algorithm 7.17 INFO-TO-DEPE
Input: An array $c[1..n(n-1)/2]$ such that for all $1 \leq i < j \leq n$, $c[k] = \chi(c_{ij})$, where $k = j(j-1)/2 - i + 1$, and a vector $b = \chi(B)$.
Output: $M = \{\chi(L) \mid L \in \mathcal{M}(\mathcal{L}_{(\mathcal{D}_{\mathcal{S}})})\}$

1. Start with $MF := \{c[k]' \mid 1 \leq k \leq n(n-1)/2\}$.
2. For all $v \in MF$, if $v = \bigwedge\{w \in MF \mid v < w\}$, then $MF := MF - \{v\}$
3. Output MF.

The complexity of the algorithm is $O(n^4 m)$

Example 7.18 The discernibility matrix of Example 5.5 can be represented as an array $c[1..6]$ in which

$$c[1] = \chi(c_{12}) = (0,1,0,1), \quad c[2] = \chi(c_{23}) = (1,0,0,1),$$
$$c[3] = \chi(c_{13}) = (1,1,0,1),$$
$$c[4] = \chi(c_{34}) = (1,0,1,1), \quad c[5] = \chi(c_{24}) = (0,0,1,0),$$
$$c[6] = \chi(c_{14}) = (0,1,1,1).$$

After Step 1, $MF = \{(1,0,1,0), (0,1,1,0), (0,0,1,0), (0,1,0,0), (1,1,0,1), (1,0,0,0)\}$. Now

- $(1,0,1,0) \neq \bigwedge\{w \in MF \mid (1,0,1,0) < w\} = \bigwedge \emptyset = \mathbf{1}_m$.
- $(0,1,1,0) \neq \bigwedge \emptyset = \mathbf{1}_m$.
- $(0,0,1,0) = \bigwedge\{(1,0,1,0),(0,1,1,0)\}$.
- $(0,1,0,0) = \bigwedge\{(0,1,1,0),(1,1,0,1)\}$.
- $(1,1,0,1) \neq \bigwedge \emptyset = \mathbf{1}_m$.
- $(1,0,0,0) = \bigwedge\{(1,0,1,0),(1,1,0,1)\}$.

Hence, $\mathcal{D}_{\mathcal{S}}$ can be represented as the set $\{(1,0,1,0), (0,1,1,0), (1,1,0,1)\}$ of vectors.

Bibliography

[1] J. C. BIOCH, T. IBARAKI, *Complexity of identification and dualization of positive Boolean functions*, Information and Computation 123 (1995), 50–63.

[2] S. BURRIS, H. P. SANKAPPANAVAR, *A course in universal algebra*, Graduate texts in mathematics, Springer–Verlag, New York, Heidelberg, Berlin, 1981.

[3] P. M. COHN, *Universal algebra*, Harper and Row, New York, 1965.

[4] B. A. DAVEY, H. A. PRIESTLEY, *Introduction to lattices and order*, Cambridge University Press, Cambridge, 1990.

[5] F. GÉCSEG AND H. JÜRGENSEN, *Algebras with dimension*, Algebra Universalis 30 (1993), 422–446.

[6] K. GŁAZEK, *Some old and new problems in the independence theory*, Colloquium Mathematicum XLII (1979), 127–189.

[7] G. GRÄTZER, *Lattice Theory: first concepts and distributive lattices*, W. H. Freeman and company, San Francisco, 1971.

[8] J. JÄRVINEN, *A representation of dependence spaces and some basic algorithms*, Fundamenta Informaticae (1996) (to appear).

[9] S. MUROGA, *Threshold logic and its applications*, Wiley–Interscience, New York, 1971.

[10] J. NOVOTNÝ, M. NOVOTNÝ , *Notes on the algebraic approach to dependence in information systems*, Fundamenta Informaticae 16 (1992), 263–273.

[11] J. NOVOTNÝ, M. NOVOTNÝ , *On dependence in Wille's contexts*, Fundamenta Informaticae 19 (1993), 343–353.

[12] M. NOVOTNÝ, Z. PAWLAK, *Algebraic theory of independence in information systems*, Fundamenta Informaticae 14 (1991), 454–476.

[13] M. NOVOTNÝ, Z. PAWLAK, *On a problem concerning dependence spaces*, Fundamenta Informaticae 16 (1992), 275–287.

[14] M. NOVOTNÝ, *Dependence spaces of information systems*, 1993 (a manuscript).

[15] E. ORŁOWSKA, Z. PAWLAK, *Representation of nondeterministic information*, Theoretical computer science 29 (1984), 27–39.

[16] E. ORŁOWSKA, *Kripke models with relative accessibility and their applications to inferences from incomplete information* In: G. MIRKOWSKA, H. RASIOWA, *Mathematical Problems in Computation Theory*, Banach Center Publications 21 (1988), 329–339.

[17] E. ORŁOWSKA, *Information Algebras*, 1995 (manuscript).

[18] C. H. PAPADIMITRIOU, *Computational Complexity*, Addison-Wesley Publishing Company, Inc., USA, 1994.

[19] Z. PAWLAK, *Information systems – theoretical foundations*, Informations Systems, Vol. 6, No. 3 (1981), 205–218.

[20] Z. PAWLAK, *Rough sets*, International Journal of Computer and Information Sciences, Vol. 11, No. 5 (1982), 341–356.

[21] Z. PAWLAK, *Rough sets. Theoretical aspects of reasoning about data*, Kluwer Academic Publishers, Dordrecht, 1991.

[22] J. A. POMYKAŁA, *On definability in the nondeterministic information system*, Bulletin of the Polish Academy of Sciences, Mathematics, Vol. 36 (1988), 193–210.

[23] J.A. POMYKAŁA, *Some remarks of approximation*, Demonstratio Mathematica, Vol. XXIV (1991), 95–104.

[24] C. M. RAUSZER, *An equivalence between indiscernibility relations in information systems and a fragment of intuitionistic logic*, Computation Theory, Lecture Notes in Computer Science, no. 208, (ed. A. SKOWRON), Spinger–Verlag, New York (1985), 298–317.

[25] C. M. RAUSZER, *Reducts in information systems*, Fundamenta Informaticae 15 (1991), 1–12.

[26] A. SKOWRON, C. RAUSZER, *The discernibility matrices and functions in information systems*. Intelligent decision support, Handbook of applications and advances of the rough set theory (ed. R. SLOWINSKI), Kluwer academic publisher, Dordrecth (1991), 331–362.

[27] M. STEINBY, *Karkeat joukot ja epätäydellinen tieto*, 1994 (a manuscript in Finnish, to appear).

[28] D. VAKARELOV, *Consequence relations and information systems*. Intelligent decision support, Handbook of applications and advances of the rough set theory (ed. R. SLOWINSKI), Kluwer academic publisher, Dordrecht (1991), 391–399.