Concept Lattices Constrained by Attribute Dependencies

Radim Bělohlávek, Vladimír Sklenář, Jiří Zacpal

Dept. Computer Science, Palacký University, Tomkova 40, CZ-779 00, Olomouc, Czech Republic, radim.belohlavek@upol.cz

Abstract. The input data to formal concept analysis consist of a collection of objects, a collection of attributes, and a table describing a relationship between objects and attributes (so-called formal context). Very often, there is an additional information about the objects and/or attributes available. In the analysis of the data, the additional information should be taken into account.

We consider a particular form of the additional information. The information is in the form of particular attribute dependencies. The primary interpretation of the dependencies is to express a kind of relative importance of attributes. We introduce the notion of a formal concept compatible with the attribute dependencies. The main gain of considering only compatible formal concepts and disregarding formal concepts which are not compatible is the reduction of the number of resulting formal concepts. This leads to a more comprehensible structure of formal concepts (clusters) extracted from the input data. We illustrate our approach by examples.

Keywords: formal context, formal concept, concept lattice, clustering, constraint, attribute dependency

1 Introduction and problem setting

Finding interesting patterns in data has traditionally been a challenging problem. Particular attention has been paid to discovering interesting clusters in data. Recently, there has been a growing interest in so-called formal concept analysis (FCA) [4] which provides methods for finding patterns and dependencies in data which can be run automatically. The patterns looked for are called formal concepts. Both foundations and applications (classification, software (re)engineering, document and text organization, etc.) of formal concept analysis are documented (see [4] and [1], and the references therein).

The central notion of all clustering methods is that of a cluster. Clusters are supposed to be meaningful pieces of data which are cohesive in some way. To have a good notion of a cluster, one should exploit all the information about the data available which can contribute to identification of meaningful clusters.

Formal concept analysis deals with input data in the form of a table with rows corresponding to objects and columns corresponding to attributes which

© V. Snášel, J. Pokorný, K. Richta (Eds.): Dateso 2004, pp. 63–73, ISBN 80-248-0457-3.
VŠB – Technical University of Ostrava, Dept. of Computer Science, 2004.

describes a relationship between the objects and attributes. The data table is formally represented by a so-called formal context which is a triplet $\langle X, Y, I\rangle$ where I is a binary relation between X and Y, $\langle x, y\rangle \in I$ meaning that the object x has the attribute y. For each $A \subseteq X$ denote by $A^{\uparrow}$ a subset of Y defined by

$$A^{\uparrow} = \{y \mid \text{for each } x \in X : \ \langle x, y\rangle \in I\}.$$

Similarly, for $B \subseteq Y$ denote by $B^{\downarrow}$ a subset of X defined by

$$B^{\downarrow} = \{x \mid \text{for each } y \in Y : \ \langle x, y\rangle \in I\}.$$

That is, $A^{\uparrow}$ is the set of all attributes from Y shared by all objects from A (and similarly for $B^{\downarrow}$). A formal concept in $\langle X, Y, I\rangle$ is a pair $\langle A, B\rangle$ of $A \subseteq X$ and $B \subseteq Y$ satisfying $A^{\uparrow} = B$ and $B^{\downarrow} = A$. That is, a formal concept consists of a set A of objects which fall under the concept and a set B of attributes which fall under the concept such that A is the set of all objects sharing all attributes from B and, conversely, B is the collection of all attributes from Y shared by all objects from A. This definition formalizes the traditional approach to concepts which is due to Port-Royal logic [2]. The sets A and B are called the extent and the intent of the concept $\langle A, B\rangle$, respectively. The set $\mathcal{B}(X, Y, I) = \{\langle A, B\rangle \mid A^{\uparrow} = B, B^{\downarrow} = A\}$ of all formal concepts in $\langle X, Y, I\rangle$ can be naturally equipped with a partial order $\leq$ defined by

$$\langle A_1, B_1\rangle \leq \langle A_2, B_2\rangle \text{ iff } A_1 \subseteq A_2 \text{ (or, equivalently, } B_2 \subseteq B_1).$$

That is, $\langle A_1, B_1\rangle \leq \langle A_2, B_2\rangle$ means that each object from A_1 belongs to A_2 (or, equivalently, each attribute from B_2 belongs to B_1). Therefore, $\leq$ models the natural subconcept-superconcept hierarchy under which dog is a subconcept of mammal.

The structure of $\mathcal{B}(X, Y, I)$ is described by the so-called main theorem of concept lattices [4,6].

Theorem 1. *(1) The set $\mathcal{B}(X, Y, I)$ is under $\leq$ a complete lattice where the infima and suprema are given by*

$$\bigwedge_{j\in J} \langle A_j, B_j\rangle = \langle \bigcap_{j\in J} A_j, (\bigcup_{j\in J} B_j)^{\downarrow\uparrow}\rangle \ , \bigvee_{j\in J} \langle A_j, B_j\rangle = \langle (\bigcup_{j\in J} A_j)^{\uparrow\downarrow}, \bigcap_{j\in J} B_j\rangle \ . \quad (1)$$

(2) Moreover, an arbitrary complete lattice $\mathbf{V} = \langle V, \leq\rangle$ is isomorphic to $\mathcal{B}(X, Y, I)$ iff there are mappings $\gamma : X \to V$, $\mu : Y \to V$ such that

(i) $\gamma(X)$ is $\bigvee$-dense in V, $\mu(Y)$ is $\bigwedge$-dense in V;
(ii) $\gamma(x) \leq \mu(y)$ iff $\langle x, y\rangle \in I$.

In the basic setting of formal concept analysis, no further information except for $\langle X, Y, I\rangle$ is taken into account. However, more often than not, both the set of objects and the set of attributes are supplied with an additional information. Further processing of the input data (formal context) should therefore take the additional information into account. For example, some attributes may be relevant (or relevant to some degree) with respect to a particular kind of decisions while some may be not. When processing a respective formal context in order to get some support for the decisions in question, the attributes which are not relevant to the decision may be disregarded. In the end, this may result in a simplification of the overall processing.

In this paper, we consider additional information which has the form of formulas (so called AD-formulas) describing particular dependencies between attributes expressing their relative importance. We introduce the notion of a formal concept compatible with an AD-formula. This enables us to eliminate formal concepts which are not compatible with the information about the relative importance of attributes. An important effect of the elimination is a natural reduction of the size of the resulting conceptual structure making the structure more comprehensible. This paper extends in a natural way our previous approach [3] in that the constraints expressible by AD-formulas are more general and thus more expressive than those of [3]. Such an extension is needed, as we discuss in the text and show by examples.

2 Constraints by attribute dependencies

Basic motivation When people categorize objects by means of the object attributes, they naturally take into account the importance of attributes. Usually, attributes which are less important are not used to form large categories (clusters, concepts). Rather, less important attributes are used to make a finer categorization within a larger category. For instance, consider a collection of certain products offered on a market, e.g. home appliances. When categorizing home appliances, one may consider several attributes like price, the purpose of the appliance, the intended placement of the appliance (kitchen appliance, bathroom appliance, office appliance, etc.), power consumption, color, etc. Intuitively, when forming appliance categories, one picks the most important attributes and forms the general categories like "kitchen appliances", "office appliances", etc. Then, one may use the less important attributes (like "price $\leq$ \$10", "price between \$15–\$40", "price $>$ \$100", etc.) and form categories like "kitchen appliance with price between \$15–\$40". Within this category, one may further form finer categories distinguished by color. This pattern of forming categories follows the rule that when an attribute y is to belong to a category, the category must contain an attribute which determines a more important characteristic of the attribute (like "kitchen appliance" determines the intended placement of the appliance). This must be true for all the characteristics that are more important than y. In this sense, the category "red appliance" is not well-formed since color is considered less important than price and the category "red appliance" does not contain any information about the price. Which attributes and characteristics are considered more important depends on the particular purpose of categorization. In the above example, it may well be the case that price be considered more important that the intended placement. Therefore, the information about the relative importance of the attributes is to be supplied by an expert (the person who determines the purpose of the categorization). Once the information has been supplied, it serves as a constraint for the formation of categories. In what follows, we propose a formal approach to the treatment of the above-described constraints to formation of categories.

Constraints by attribute-dependency formulas Consider a formal context $\langle X, Y, I\rangle$. We consider constraints expressed by formulas of the form

$$y \sqsubseteq y_1 \sqcup \cdots \sqcup y_n. \tag{2}$$

Formulas of this form will be called AD-formulas (attribute-dependency formulas). The set of all AD-formulas will be denoted by ADF. Let now $\mathcal{C} \subseteq ADF$ be a set of AD-formulas.

Definition 1. A formal concept $\langle A, B\rangle$ satisfies an AD-formula (2) if we have that

$$\text{if } y \in B \text{ then } y_1 \in B \text{ or } \cdots \text{ or } y_n \in B.$$

The fact that $\langle A, B\rangle \in \mathcal{B}(X, Y, I)$ satisfies an AD-formula φ is denoted by $\langle A, B\rangle \models \varphi$. Therefore, $\models$ is the basic satisfaction relation (being a model) between the set $\mathcal{B}(X, Y, I)$ of all formal concepts (models, structures) and the set ADF of all AD-formulas (formulas).

As usual, $\models$ induces two mappings, Mod : $2^{ADF} \to 2^{\mathcal{B}(X,Y,I)}$ assigning a subset

$$\mathrm{Mod}(\mathcal{C}) = \{\langle A, B\rangle \in \mathcal{B}(X, Y, I) \mid \langle A, B\rangle \models \varphi \text{ for each } \varphi \in \mathcal{C}\}$$

to a set $\mathcal{C} \subseteq ADF$ of AD-formulas, and Fml : $2^{\mathcal{B}(X,Y,I)} \to 2^{ADF}$ assigning a subset

$$\mathrm{Fml}(U) = \{\varphi \in ADF \mid \langle A, B\rangle \models \varphi \text{ for each } \langle A, B\rangle \in U\}$$

to a subset $U \subseteq \mathcal{B}(X, Y, I)$.

The following result is immediate [5].

Theorem 2. *The mappings* Mod *and* Fml *form a Galois connection between* ADF *and* $\mathcal{B}(X, Y, I)$. *That is, we have*

$$\mathcal{C}_1 \subseteq \mathcal{C}_2 \mathit{implies} \mathrm{Mod}(\mathcal{C}_2) \subseteq \mathrm{Mod}(\mathcal{C}_1), \tag{3}$$
$$\mathcal{C} \quad \subseteq \quad \mathrm{Fml}(\mathrm{Mod}(\mathcal{C})), \tag{4}$$
$$U_1 \subseteq U_2 \mathit{implies} \mathrm{Fml}(U_2) \subseteq \mathrm{Fml}(U_1), \tag{5}$$
$$U \quad \subseteq \quad \mathrm{Mod}(\mathrm{Fml}(U)). \tag{6}$$

for any $\mathcal{C}, \mathcal{C}_1, \mathcal{C}_2 \subseteq ADF$, *and* $U, U_1, U_2 \subseteq \mathcal{B}(X, Y, I)$.

Thus, more generally, for $U \subseteq \mathcal{B}(X, Y, I)$ and $\mathcal{C} \subseteq ADF$ we write $U \models \mathcal{C}$ if $U \subseteq \mathrm{Mod}(\mathcal{C})$ which is equivalent to $\mathcal{C} \subseteq \mathrm{Fml}(U)$ (the meaning: each $\langle A, B\rangle \in U$ satisfies each $\varphi \in \mathcal{C}$).

Definition 2. For $\mathcal{C} \subseteq ADF$ we put

$$\mathcal{B}_{\mathcal{C}}\,(X, Y, I) = \mathrm{Mod}(\mathcal{C})$$

and call it the *constrained (by $\mathcal{C}$) concept lattice* induced by $\langle X, Y, I\rangle$ and $\mathcal{C}$.

For simplicity, we also denote $\mathcal{B}_{\mathcal{C}}\,(X, Y, I)$ simply by $\mathcal{B}_{\mathcal{C}}$. That is, $\mathcal{B}_{\mathcal{C}}\,(X, Y, I)$ is the collection of all formal concepts from $\mathcal{B}\,(X, Y, I)$ which satisfy each AD-formula from $\mathcal{C}$ (satisfy all constraints from $\mathcal{C}$).

Note that (3)–(6) have a natural interpretation. For instance, (3) says that the more formulas we put to $\mathcal{C}$ (the more constraints), the fewer formal concepts are in $\mathcal{B}_{\mathcal{C}}$.

Remark 1. (1) In [3], we introduced constraints by a hierarchy on Y which is represented by a partial order $\unlhd$ on Y. A formal concept $\langle A, B\rangle \in \mathcal{B}\,(X, Y, I)$ is called compatible with $\unlhd$ if for each $y \in B$ and $y \unlhd y'$ we have $y' \in B$. Denote $\mathcal{B}\,(X, \langle Y, \unlhd\rangle, I)$ the set of all formal concepts from $\mathcal{B}\,(X, Y, I)$ which are compatible with $\unlhd$. It is clear that putting $\mathcal{C}_{\unlhd} = \{y_1 \sqsubseteq y_2 \mid \langle y_1, y_2\rangle \in \unlhd\}$, we have $\mathcal{B}\,(X, \langle Y, \unlhd\rangle, I) = \mathcal{B}_{\mathcal{C}_{\unlhd}}\,(X, Y, I)$. This way our current approach generalizes that one of [3].

(2) Note that our present approach is needed. For instance, if y, y_1, and y_2 stand for “price > \$100”, “kitchen appliance”, and “office appliance”, respectively, then $y \sqsubseteq y_1 \sqcup y_2$ represents a natural constraint which cannot be directly expresses by a hierarchy $\unlhd$ in the sense of [3].

In the rest of this section we briefly discuss selected topics related to constraints by AD-formulas. Due to the limited scope, we omit details.

Structure of $\mathcal{B}_{\mathcal{C}}\,(X, Y, I)$ Contrary to [3], we lose some nice properties under the present approach. For example, although $\mathcal{B}_{\mathcal{C}}\,(X, Y, I)$ is a partially ordered subset of $\mathcal{B}\,(X, Y, I)$, it does no need not be a sup-sublattice of $\mathcal{B}\,(X, Y, I)$ as is the case of $\unlhd$.

Example 1. Let $X = \{x_1, x_2\}$, $Y = \{y_1, y_2, y_3\}$, $I = \{\langle x_1, y_2\rangle, \langle x_1, y_3\rangle, \langle x_2, y_1\rangle, \langle x_2, y_3\rangle\}$, $\mathcal{C} = \{y_3 \sqsubseteq y_1 \sqcup y_2\}$. Then $\mathcal{B}_{\mathcal{C}}$ is not a sup-sublattice of $\mathcal{B}\,(X, Y, I)$.

Entailment of AD-formulas Another interesting issue, in fact, a very important one, is that of entailment of AD-formulas (i.e. the notion of entailment of an AD-formula by a set of AD-formulas). Namely, AD-formulas which follow from $\mathcal{C}$ may be ignored because do not represent any additional constraint. Conversely, it might be interesting to look for a base of a set $\mathcal{C}$ of AD-formulas, i.e. a subset $\mathcal{C}' \subseteq \mathcal{C}$ such that each $\varphi \in \mathcal{C}$ follows from $\mathcal{C}'$ and $\mathcal{C}'$ is a minimal one with this property. Due to the limited scope, we omit any further details.

Expressive power of AD-formulas A given $y \in Y$ may occur on left hand-side of several AD-formulas. For example, we may have $y \sqsubseteq y_1 \sqcup y_2$ and $y \sqsubseteq y_3 \sqcup y_4$. Then, for a formal concept $\langle A, B\rangle$ to be compatible, it has to satisfy the following: whenever $y \in B$ then it must be the case that $y_1 \in B$ or $y_2 \in B$, and $y_3 \in B$ or $y_4 \in B$. Therefore, it is tempting to allow for expressions of the form

$$y \sqsubseteq (y_1 \sqcup y_2) \sqcap (y_3 \sqcup y_4)$$

with the intuitively clear meaning of compatibility of a formal concept and a formula of this generalized form. Note that a particular form is also e.g. $y \sqsubseteq y_2 \sqcap y_3$. One may also want to extend this form to formulas containing disjunctions of conjunctions, e.g.

$$y \sqsubseteq (y_1 \sqcap y_2) \sqcup (y_3 \sqcap y_4).$$

It is not difficult, however, somewhat tedious, to show that the expressive power of such generalized formulas remains the same. More precisely, to each set $\mathcal{C}$ of generalized formulas there exists a set $\mathcal{C}'$ of ordinary AD-formulas such that for each formal concept $\langle A, B\rangle$ we have that $\langle A, B\rangle \models \mathcal{C}$ iff $\langle A, B\rangle \models \mathcal{C}'$.

3 Examples

We now present illustrative examples. We assume that the reader is familiar with Hasse diagrams which will be used for visualization of concept lattices and attribute hierarchies. We label the nodes corresponding to formal concepts by boxes containing concept descriptions. For example, $(\{1, 3, 7\}, \{3, 4\})$ is a description of a concept the extent of which consists of objects 1, 3, and 7, and the intent of which consists of attributes 3 and 4.

Example 2. **Using attribute dependencies for generation of views on databases.** Suppose we have a relational database with particular car models as objects and selected car properties as attributes. We have attributes like “hatchback”, “sedan”, “diesel engine”, “gasoline engine”, “air-conditioning”, “ABS”, etc. This data can be understood as a (bivalent) formal context. This context induces a corresponding concept lattice containing all formal concepts hidden in the database. In general, this concept lattice contains a large number of formal concepts. This fact makes the concept lattice not comprehensible by humans. With respect to a particular aim (e.g. a decision making), the concept lattice will contain both important and natural concepts as well as concepts which are considered not important.

To get a more precise idea, suppose a customer wants to buy a car and wants to look at the concept lattice to help him select one. He has a certain idea of what the car should fulfill. Some car properties can be more important for him then others.

Consider the formal context $\langle X, Y, I\rangle$ in Tab. 1 . The context contains cars as the objects (labeled 1–8) and some of their properties as the attributes (labeled

	1	2	3	4	5	6	7	8
car 1	1	0	1	0	0	1	0	1
car 2	1	0	1	0	1	1	0	1
car 3	0	1	1	0	0	0	0	1
car 4	0	1	0	1	1	0	0	0
car 5	0	1	1	0	1	1	0	0
car 6	0	1	0	1	0	1	1	0
car 7	0	1	0	1	1	1	1	1
car 8	0	1	0	1	0	0	0	1

attributes: 1 - diesel engine, 2 - gasoline engine, 3 - sedan, 4 - hatchback, 5 - air-conditioning, 6 - airbag, 7 - power stearing, 8 - ABS

Table 1. Formal context given by cars and their properties.

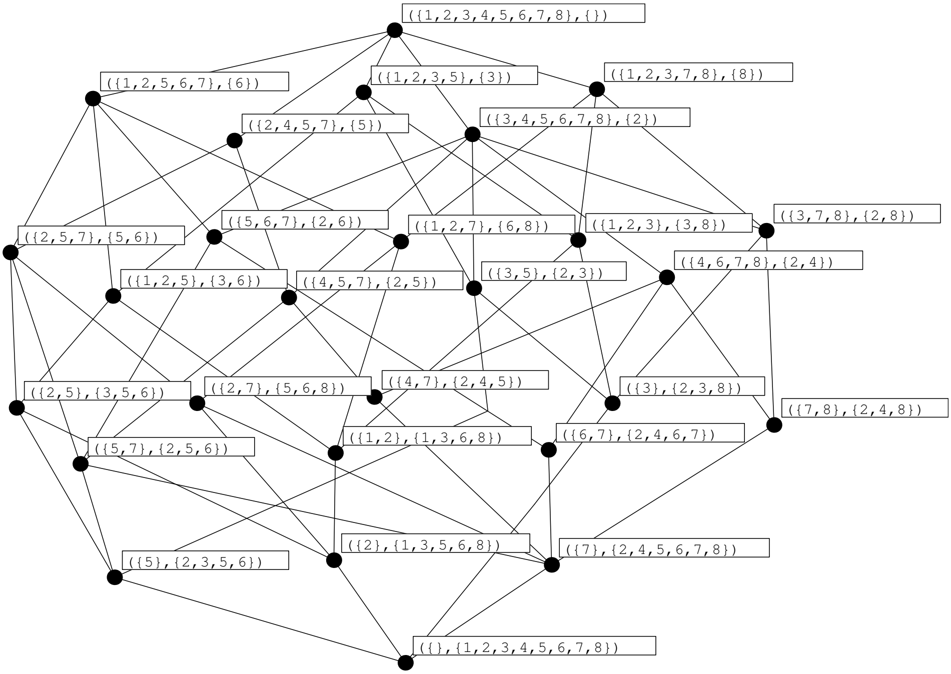

Fig. 1. Concept lattice corresponding to the context from Tab. 1

1–8). The concept lattice $\mathcal{B}(X, Y, I)$ corresponding to formal concept $\langle X, Y, I\rangle$ contains 27 formal concepts and is depicted in Fig. 1.

The formal concepts of $\mathcal{B}(X, Y, I)$ represent all concept-clusters that are present in the data. No attention is paid to importance or relative importance of attributes.

Let us now consider some attribute dependencies and the corresponding constrained concept lattices $\mathcal{B}(X, Y, I)$.

First, consider a set of AD-formulas (7)–(12). They represent the fact that most important car propeties (for a particular user) are the kind of engine, etc.

$$air - conditioning \sqsubseteq hatchback \sqcup sedan \tag{7}$$
$$powerstearing \sqsubseteq hatchback \sqcup sedan \tag{8}$$
$$airbag \sqsubseteq hatchback \sqcup sedan \tag{9}$$
$$ABS \sqsubseteq hatchback \sqcup sedan \tag{10}$$
$$hatchback \sqsubseteq gasoline\ engine \sqcup diesel\ engine \tag{11}$$
$$sedan \sqsubseteq gasoline\ engine \sqcup diesel\ engine \tag{12}$$

The concept lattice $\mathcal{B}(X, Y, I)$ constrained by AD-formulas (7)–(12) contains 13 formal concepts and is depicted in Fig. 2.

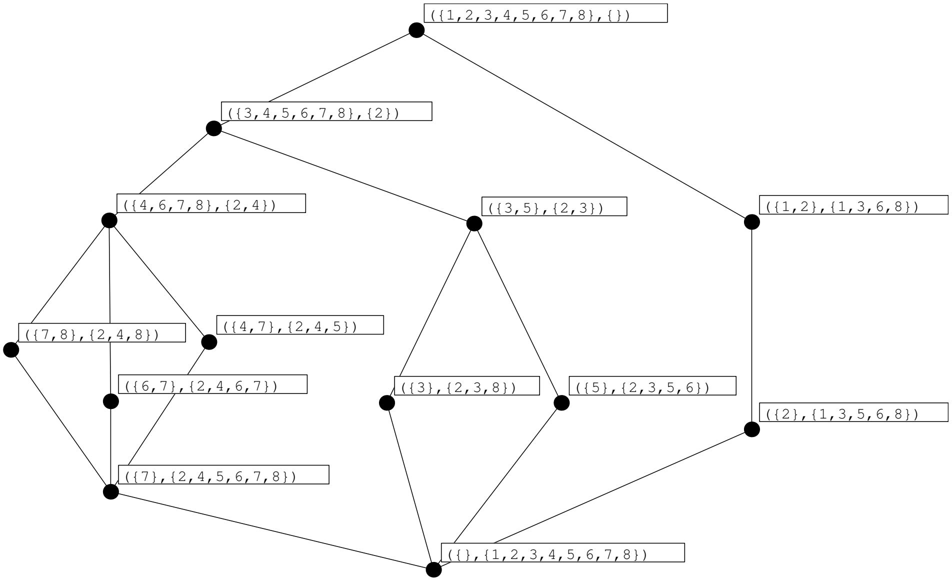

Fig. 2. Concept lattice constrained by AD-formulas (7)–(12)

Second, consider a set of AD-formulas (13)–(18). Contrary to the previous example, the importance of the type of a car and the kind of the engine are reversed.

$$air - conditioning \sqsubseteq diesel\ engine \sqcup gasoline\ engine \tag{13}$$

$$powerstearing \sqsubseteq diesel\ engine \sqcup gasoline\ engine \tag{14}$$

$$airbag \sqsubseteq diesel\ engine \sqcup gasoline\ engine \tag{15}$$

$$ABS \sqsubseteq diesel\ engine \sqcup gasoline\ engine \tag{16}$$

$$gasoline\ engine \sqsubseteq hatchback \sqcup sedan \tag{17}$$

$$diesel\ engine \sqsubseteq hatchback \sqcup sedan \tag{18}$$

The concept lattice $\mathcal{B}(X, Y, I)$ constrained by AD-formulas (13)–(18) contains 14 formal concepts and is depicted in Fig. 3.

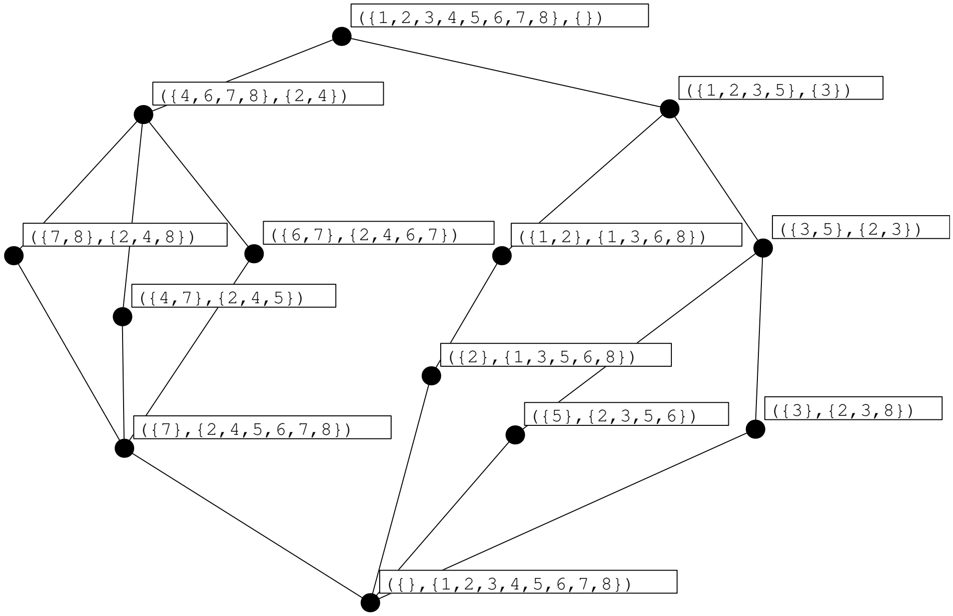

Fig. 3. Concept lattice constrained by AD-formulas (13)–(18)

Third, suppose the user finds the most important car property to be safety. The situation is described by AD-formulas (19)–(26)

$$air - conditioning \sqsubseteq diesel\ engine \sqcup gasoline\ engine \tag{19}$$

$$powerstearing \sqsubseteq diesel\ engine \sqcup gasoline\ engine \tag{20}$$

$$gasoline\ engine \sqsubseteq hatchback \sqcup sedan \tag{21}$$

$$diesel\ engine \sqsubseteq hatchback \sqcup sedan \tag{22}$$

$$sedan \sqsubseteq ABS \tag{23}$$

$$hatcback \sqsubseteq ABS \tag{24}$$

$$ABS \sqsubseteq airbag \tag{25}$$

$$airbag \sqsubseteq ABS \tag{26}$$

The concept lattice $\mathcal{B}(X, Y, I)$ constrained by AD-formulas (19)–(26) contains 6 formal concepts and is depicted in Fig. 4.

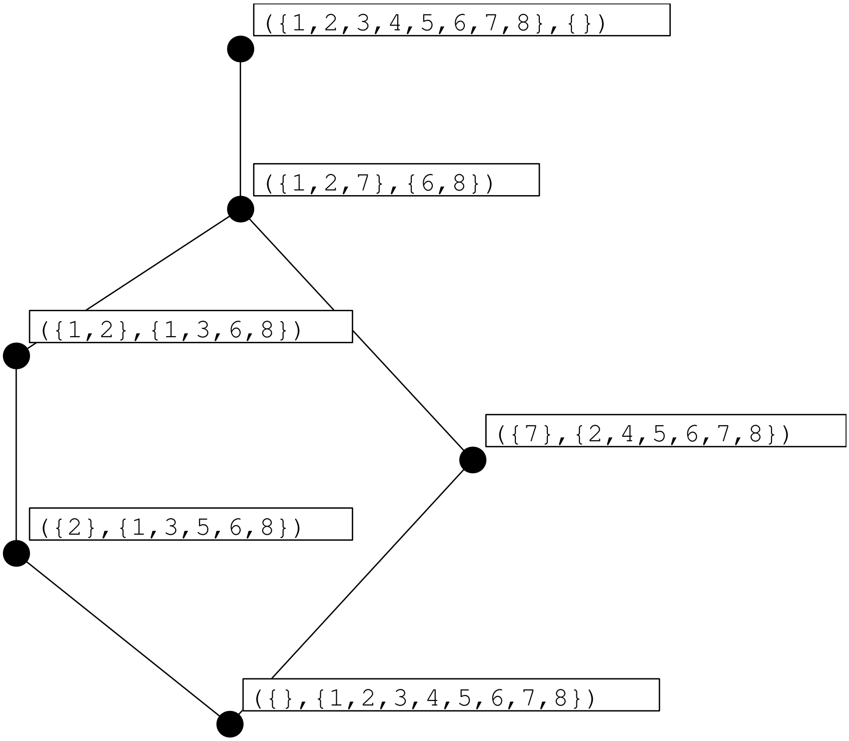

Fig. 4. Concept lattice constrained by AD-formulas (19)–(26)

Acknowledgement The research of the first author was supported by grant No. 201/02/P076 of the GAČR.

References

1. http://www.mathematik.tu-darmstadt.de/ags/ag1/Literatur/literatur_de.html
2. Arnauld A., Nicole P.: *La logique ou l'art de penser.* 1662. Also in German: *Die Logik oder die Kunst des Denkens.* Darmstadt, 1972.
3. Bělohlávek R., Sklenář V., Zacpal J.: Formal concept analysis with hierarchically ordered attributes. *Int. J. General Sytems* (to appear).
4. Ganter B., Wille R.: *Formal concept analysis. Mathematical Foundations.* Springer-Verlag, Berlin, 1999.
5. Ore O.: Galois connections. *Trans. Amer. Math. Soc.* **55**(1944), 493–513.
6. Wille R.: Restructuring lattice theory: an approach based on hierarchies of concepts. In: Rival I.: *Ordered Sets.* Reidel, Dordrecht, Boston, 1982, 445—470.