Foundations

Note on representing attribute reduction and concepts in concept lattice using graphs

Jan Konecný

Published online: 29 April 2019
© Springer-Verlag GmbH Germany, part of Springer Nature 2019

Abstract
Mao (Soft Computing 21(24):7293–7311, 2017) claims to make contributions to the study of reduction in attributes in concept lattices by using graph theory. Some of her results could affect development of formal concept analysis; in particular, her three new algorithms for formal concept enumeration promise a lesser time complexity. Unfortunately, as we show, all the three algorithms are incorrect.

Keywords Clarification · Reduction · Formal context · Graph · Formal concept enumeration

1 Introduction
In her recent paper, Mao (2017) made some important claims which could have significant impact on the further development of formal concept analysis (FCA) and related fields: specifically a new method of formal concept enumeration with superior time complexity is proposed. The method is represented by three algorithms, each enumerating specific portion of the concept. Regrettably, it turns out that the algorithms are incorrect. We demonstrate, using simple examples, that they fail to output some concept or that they output them multiple times.

2 Preliminaries
We use the same notions and the same notations as Mao (2017); however, we need to also recall a few notions from (Ganter and Wille 1999) to explain our case. Thus, for the reader’s convenience, we provide full preliminaries.

An input to FCA is a triplet \((O, P, I)\), called a formal context, where \(O\) and \(P\) are finite non-empty sets of objects and attributes, respectively, and \(I\) is a binary relation between \(O\) and \(P\); \((o, a) \in I\) means that the object \(o\) has the attribute \(a\). Finite formal contexts are usually depicted as tables, in which rows represent objects, columns represent attributes, and each entry contains a cross if the corresponding object has the corresponding attribute, and is otherwise left blank (see top parts of Figs. 3, 4, and 5 for example).

The formal context induces the following operators:

\[ \uparrow : 2^O \to 2^P \] assigns to a set \(X\) of objects the set \(X \uparrow\) of all attributes shared by all the objects in \(X\).

\[ \downarrow : 2^P \to 2^O \] assigns to a set \(B\) of attributes the set \(B \downarrow\) of all objects which share all the attributes in \(B\).

For singletons, we use shortened notation and write \(a \uparrow\), \(a \downarrow\) instead of \(\{a\} \uparrow\), \(\{a\} \downarrow\), respectively.

Formal concept is a pair \((X, B)\) of sets \(X \subseteq O, B \subseteq P\), s.t. \(X \uparrow = B\) and \(B \downarrow = X\). The first component of a formal concept is called extent; the second one is called intent. The collection of all formal concepts in \((O, P, I)\) is denoted \(\mathcal{B}(O, P, I)\). The collection \(\mathcal{B}(O, P, I)\) with order \(\leq\) defined by \((X_1, B_1) \leq (X_2, B_2)\) if \(X_1 \subseteq X_2\) for all formal concepts \((X_1, B_1), (X_2, B_2) \in \mathcal{B}(O, P, I)\), forms a complete lattice called a concept lattice.

We call a computation of \(\mathcal{B}(O, P, I)\) a formal concept enumeration.

We consider the following two binary relations on \(P\) induced by a formal context:

- dependency \(\subseteq\), defined by \(a \subseteq b\) if \(a \downarrow \subseteq b \downarrow\) for all \(a, b \in P\).
- equivalence \(\equiv\), defined by \(a \equiv b\) if \(a \downarrow = b \downarrow\) for all \(a, b \in P\).

Communicated by A. Di Nola.

Jan Konecný
jan.konecny@upol.cz

1 Department of Computer Science, Palacky University, Olomouc, 17. listopadu 12, 77146 Olomouc, Czech Republic
Formal context \((O, P, I)\) is called clarified if \(a_1 \equiv a_2\) implies \(a_1 \equiv a_2\) for any \(a_1, a_2 \in P\), i.e. if it does not contain duplicate columns. Removal of duplicate columns (keeping one representative of each \(\equiv\)-class) is called a clarification.

An attribute \(a\) is called reducible in formal context \((O, P, I)\) if there is \(Z \subseteq P \setminus \{a\}\) such that \(a^\perp = Z^\perp\) (equivalently, if \(\mathcal{B}(O, P, I)\) and \(\mathcal{B}(O, P\setminus\{a\}, I \cap (O \times P\setminus\{a\}))\) are isomorphic). Formal context is reduced if it has no reducible attributes. Removal of reducible attributes is called a reduction. Ganter and Wille (1999) provide efficient method for clarification and reduction.

Originally for rough set theory, Pawlak (1982) proposed three types of attributes. These types were introduced into FCA by Zhang et al. (2005). We omit their definitions and just explain their relationship to the reducible and irreducible attributes: absolutely necessary (core) attributes are exactly irreducible attributes; relatively necessary attributes are those reducible attributes that become irreducible when their duplicates are removed; and absolutely unnecessary attributes are those which are neither absolutely nor relatively necessary.

3 The use of graph theory is trivial and the results are not novel

Mao (2017) claims to propose a method of attribute reduction which utilizes graph theory and is based on removing vertices from a particular graph; this should distinguish her method from other known approaches (Ganter and Wille 1999; Zhang et al. 2005) which are based on the removal of attributes.

Specifically, she introduces a so-called pre-weighted relevant graph \(G(O, P, I)\) (derived from a formal context \((O, P, I)\)) as a graph with vertices being the attributes in \(P\) and edges being given as follows: for \(a, b \in P\) and \(a \neq b\),

- if \(x^\perp = y^\perp\) there is a bi-arc joining \(x\) and \(y\), i.e. \(x \Leftrightarrow y\),
- if \(x^\perp \subseteq y^\perp\) there is an arc joining \(x\) to \(y\), i.e. \(x \rightarrow y\).

Thus, \(G(O, P, I)\) is basically \((O, P, I)\) with the relations \(\subseteq\) and \(\equiv\) (specifically, the arcs \(\rightarrow\) correspond to pairs in \(\subseteq\) and \(\equiv\) and the bi-arcs \(\Leftrightarrow\) correspond to non-reflexive pairs in \(\equiv\)). This represents the entire utilization of graph theory in Mao’s work.

Mao describes the attribute reduction in three steps (see Fig. 1):

1. For an input formal context \((O_0, P_0, I_0)\) satisfying

\[ o^\perp \neq \emptyset \text{ and } a^\perp \neq \emptyset \text{ for all } o \in O_0, a \in P_0 \]

the formal context \((O_1, P_1, I_1)\) is found by removing full-row objects and full-column attributes from \((O_0, P_0, I_0)\).

2. For \((O_1, P_1, I_1)\) find context \((O_2, P_2, I_2)\) by clarification.

3. For \((O_2, P_2, I_2)\) find context \((O_3, P_3, I_3)\) by reduction.

Fig. 1 Three-phase process of attribute reduction in Mao (2017)

- First, the relations are used the same way as in Ganter and Wille (1999) (where they are not pre-computed), or they present a trivial improvement. As an example of the latter, Mao (2017, Theorem 3.1) characterizes objects with full row. The theorem states that we need not check whether the object has all attributes; instead, we can just check whether it has all attributes that are minimal w.r.t. \(\subseteq\).
- Second, if the complexity is taken into account, it should be said that the computation of \(\equiv\) and \(\subseteq\) for \((O, P, I)\) requires the same time as entire clarification and reduction using classic methods described in basic literature (Ganter and Wille 1999). Mao does not mention the complexity of the computation of \(\equiv\) and \(\subseteq\) neither does she refer to the classic methods.

Substitution of the basic notions with the equivalent newly introduced notions is Mao’s main resource for results. Instead of removal attributes from \((O, P, I)\), she removes the corresponding vertices from \(G(O, P, I)\). Rephrased, using basic notions of FCA, the results become trivial or are already well known.

Specifically, Theorem 3.2 states that duplicate attributes are reducible, which is obvious. Theorem 3.3 states that in the clarified context \((O_2, P_2, I_2)\) no attributes are relatively necessary, which is obvious because \((O_2, P_2, I_2)\) is obtained by clarification of the context \((O_1, P_1, I_1)\). Theorem 3.4 is an overcomplicated characterization of reducible attributes, which after a straightforward simplification becomes the one.
by Ganter and Wille (1999). Theorem 3.5 states that in the reduced context \((O_3, P_3, I_3)\) all attributes are irreducible, which again is obvious, because \((O_3, P_3, I_3)\) is obtained by reduction of \((O_2, P_2, I_2)\). Finally, Theorems 3.6 and 3.7 explain trivial relationships between Pawlak’s types of attributes in \(P_0, P_1, P_2,\) and \(P_3\).

### 4 The proposed algorithms are incorrect

Mao (2017) proposes a method of computing formal concepts in \((O, P, I)\); more specifically, the subset \(\mathcal{A}\) contains all formal concepts excluding the top and the bottom concepts of \(B(O, P, I)\) and the attribute concepts. The method consists of three algorithms which each compute different portions of \(\mathcal{A}\) with the aid of a pre-weighted relevant graph; their input is a pre-weighted relevant graph and a maximal attribute \(c_1\).

1. Algorithm 1 computes the set \(\mathcal{F}\) containing those concepts \((A, B) \in \mathcal{A}\) which satisfy \(c_1 \in B\) and \(b \notin B\) for all \(b \in N^{+}_{G(O,P,I)}(c_1)\) (see Fig. 2 (top)).
2. Algorithm 2 computes the set \(\mathcal{S}\) containing those concepts \((A, B) \in \mathcal{A}\) which satisfy \(c_1 \in B\) and \(b \in B\) for some \(b \in N^{+}_{G(O,P,I)}(c_1)\) (see Fig. 2 (middle)).
3. Algorithm 3 computes the set \(\mathcal{T}\) containing those concepts \((A, B) \in \mathcal{A}\) which satisfy \(c_1 \notin B\) (see Fig. 2 (bottom)).

\(N^{+}_{G(O,P,I)}(c_1)\) in items 1. and 2. denotes the lower cone of attribute \(c_1\) w.r.t. \(\sqsubseteq\), excluding \(c_1\) itself.

All three algorithms are described in a very complicated way which makes them almost unreadable. More importantly, they are incorrect. In what follows, we present the examples in which the algorithms fail to deliver correct outputs.

We need to recall some additional notation from (Mao 2017):

- \(V^{\rightarrow a}\) denotes the upper cone of attribute \(a\) w.r.t. \(\sqsubseteq\), excluding \(a\) itself;
- \(\omega(a)\) denotes the pre-weight of \(a\), i.e. \(\omega(a) = a^\downarrow\).

### 4.1 Example for Algorithm 1

Consider the formal context depicted in Fig. 3 (top).

We demonstrate that algorithm 1 fails to deliver correct output for its pre-weighted relevant graph (Fig. 3 (bottom)) and attribute \(c_1\). We have \(\mathcal{C} = \{c_1, c_2\}\) and \(N^{+}_{G(O,P,I)}(c_1) = \{b, n\}\). The following represents how the algorithm runs when we exactly follow exactly the steps in Mao (2017).

\[(step 1)\]

\(H_1 = \{x \in P \setminus \{c_1\} \mid x \notin N^{+}_{G(O,P,I)}(c_1)\}\) and \(\omega(c_1) \cap \omega(x) \neq \emptyset\) = \(\{c_2\}\).
As $H_1 \neq \emptyset$ is the case, we select $h_1 \in H_1$. There is only one option; thus, we set $h_1 := c_2$. We compute extent $A_1$ and $B_1$ as

$$A_1 = \omega(c_1) \cap \omega(c_2) = [3] \quad \text{and} \quad B_1 = \{c_1, c_2\}. $$

As $H_1 \setminus B_1 = \emptyset$, the condition of non-existence of $h \in H_1 \setminus B_1$ with $\omega(c_1) \cap \omega(c_2) \subset \omega(h)$ is trivially satisfied, and $(A_1, B_1)$ is outputted.

(step 2) As $|H_1| = 1$, we stop the computation.

We obtained a pair $([3], \{c_1, c_2\})$ as an output, but this pair is not a formal concept. If we close it, we obtain $([3], \{c_1, c_2, b\})$ which should not be in $F$, since it contains $b$.

### 4.2 Example for Algorithm 2

Consider the formal context depicted in Fig. 4 (top).

We demonstrate that algorithm 2 fails to deliver the correct output for its pre-weighted relevant graph (Fig. 4 (bottom)) and attribute $c_1$. We have that $C = \{c_1, c_2, c_3\}$ and $N_{G(O,P,I)}^+(c_1) = \{b\}$.

(step 6) We set $b_1 := b$ as it is the only option, and we compute$$H_{b_1} = \{c_2, c_3\}. $$

(step 7) We have $H_{b_1} \neq \emptyset$. We select $d_1 := c_2$. We compute

$$A_{b_1} := \omega(b_1) \cap \omega(d_1) = \omega(b) \cap \omega(c_2) = [3],$$

$$B_{b_1} := \{b_1, d_1\} \cup V_3^{-b_1} \cup V_3^{-d_1} = \{b, c_2\} \cup \{c_1\} \cup \emptyset = \{b, c_1, c_2\}. $$

Since $A_{b_1}$ is empty, the algorithm stops.

The algorithm terminated without outputting a formal concept $([4], \{c_1, c_3, b\})$, which belongs to $S$.

### 4.3 Example for Algorithm 3

The third algorithm uses the previous two algorithms. Even if algorithm 1 and algorithm 2 were correct, algorithm 3 would be still incorrect, and this is shown for the formal context depicted in Fig. 5 (top), its pre-weighted relevant graph depicted in Fig. 5 (bottom), and its attribute $c_1$.

(step 15) $C := C \setminus \{c_1\} = \{c_2, c_3\};$

$$G(O, P, I) := G(O, P, I) \setminus (c_1 \cup N_{G(O,P,I)}^+(c_1));$$

i.e. $G(O, P, I)$ now contains only vertices $c_2$ and $c_3$.

(step 16) We use algorithms 1 and 2 for updated graph $G(O, P, I)$ and $c_2$. It is not clear whether (1) includes removal of $c_1$ from $P$:

- If yes, we get $([3], \{c_2, c_3\})$ as one of the outputs of algorithms 1 and 2, but it is not a formal concept.
4.4 Bogus complexity analysis

As the method of formal concept enumeration is incorrect, we could simply disregard its complexity analysis; however, even then there is a problem which needs mentioning.

In the description of both attribute reduction and formal concept enumeration, Mao uses steps which perform intersections and/or comparisons of pre-weights, i.e. subsets of the set $O$. In the accompanied complexity analysis, she claims complexity of these steps to be $O(|P|^2)$ or $O(|P|^3)$, i.e. independent of the size of the subsets or the size of $O$. This could be actually achieved if all the intersections of pre-weights were pre-computed; however, a collection of all the intersections is, in fact, the collection of all extents. As each extent uniquely determines its formal concept, this assumption means that Mao’s method of formal concept enumeration requires all formal concepts to be pre-computed.

To sum up, either the claimed complexities are incorrect, or Mao works from an assumption that makes the actual computation superfluous.

5 Conclusion

Mao (2017) studies attribute reduction and formal concept enumeration in FCA with the aid of graph theory. Some of her claims could have significant impact on the further development of FCA and related fields. Her main result are algorithms for formal concept enumeration. Unfortunately, as we showed in this paper, these algorithms are incorrect.

Acknowledgements  The work was supported by Grant No. 15-17899S, “Decompositions of Matrices with Boolean and Ordinal Data: Theory and Algorithms”, of the Czech Science Foundation.

Compliance with ethical standards

Conflict of interest  The author declares that he has no conflict of interest.

Ethical approval  This article does not contain any studies with human participants or animals performed by the author.

References

Ganter B, Wille R (1999) Formal concept analysis—mathematical foundations. Springer, Berlin
Mao H (2017) Representing attribute reduction and concepts in concept lattice using graphs. Soft Comput 21(24):7293–7311. https://doi.org/10.1007/s00500-016-2441-2
Pawlak Z (1982) Rough sets. Int J Comput Inf Sci 11(5):341–356
Zhang W, Wei L, Qi J (2005) Attribute reduction theory and approach to concept lattice. Sci China Ser F-Inf Sci 48(6):713. https://doi.org/10.1360/122004-104

Publisher’s Note  Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.