Computing and Informatics, Vol. 40, 2021, 368–386, doi: 10.31577/cai_2021_2_368

NON-REDUNDANT IMPLICATIONAL BASE OF MANY-VALUED CONTEXT USING SAT

Taufiq Hidayat

Fakulti Teknologi Maklumat dan Komunikasi (FTMK)
Universiti Teknikal Malaysia Melaka (UTeM)
Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia
&
Department of Informatics
Islamic University of Indonesia
55584 Yogyakarta, Indonesia
e-mail: p031710045@student.utem.edu.my, taufiq.hidayat@uii.ac.id

Asmala bin Ahmad, Mohammad Ishak bin Desa

Fakulti Teknologi Maklumat dan Komunikasi (FTMK)
Universiti Teknikal Malaysia Melaka (UTeM)
Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia
e-mail: {asmala, mohammad.ishak}@utem.edu.my

Abstract. Some attribute implications in an implicational base of a derived context of many-valued context can be inferred from some other attribute implications together with its scales. The scales are interpretation of some values in the many-valued context therefore they are a prior or an existing knowledge. In knowledge discovery, the such attribute implications are redundant and cannot be considered as new knowledge. Therefore the attribute implicational should be eliminated. This paper shows that the redundancy problem exists and formalizes a model to check the redundancy.

Keywords: Attribute implication, background knowledge, SAT problem

1 INTRODUCTION

Formal context is a simple data structure, which is defined as a triple (G, M, I) where G is a set of objects, M is a set of attributes, and $I \subseteq G \times M$. If $(g, m) \in I$ where $g \in G$ and $m \in M$ then (g, m) is read as "object g has attribute m" [1, 2]. Figure 1 is an example of formal context represented by a cross table. The formal context is about small natural number. In the formal context,

$$G = \{1, 2, \ldots, 10\},$$

$$M = \{odd, even, greater\ \ than\ 2, greater\ \ than\ 5, prime, square\}.$$

	odd	even	greater than 2	greater than 5	prime	square
1	×					×
2		×			×	
3	×		×		×	
4		×	×			×
5	×		×		×	
6		×	×	×		
7	×		×	×	×	
8		×	×	×		
9	×		×	×		×
10		×	×	×		

Figure 1. Formal context of small natural number

Formal context is also able to represent a data table (relational data). A data table will be represented by many-valued context. By scaling, the many-valued context will be transformed into a one-valued context [1, 2, 3]. The one-valued context is called a derived context. In this form, the many-valued context will be analyzed.

Formal Concept Analysis (FCA) is a study to extract knowledge from the formal context. The study is useful in knowledge discovery of data. Three forms of knowledge discovery offered by FCA are clusters (which are called formal concepts), data dependencies (which are called attribute implications), and visualization of formal concepts by single hierarchical diagram (which is called concept lattice) [4]. Many researches are conducted in application of formal concepts analysis to knowledge discovery [5, 6, 7, 8, 9, 10, 11].

An attribute implication of formal context (G, M, I) is an implication in a form $A \to B$ where $A, B \subseteq M$. The attribute implication means that all objects which have all attributes in A also have all attributes in B. It holds in the formal context (G, M, I) if it holds in each object $g \in G$. These following attribute implications hold in the formal context in Figure 1:

1. $\{even, square\} \Rightarrow \{greater\ than\ 2\}$,
2. $\{prime, greater\ than\ 2\} \Rightarrow \{odd\}$,
3. $\{prime, greater\ than\ 5\} \Rightarrow \{odd\}$.

A set of attribute implications is an implicational base of a formal context (G, M, I) if the attribute implications are sound, complete, and non-redundant with respect to the formal context [2]. There are some algorithms to generate an implicational base.

However, regarding the implicational base, sometimes there are attribute implications which are already known or can be inferred from other attribute implications together with our existing knowledge. We call the existing knowledge as background knowledge. This following simple example illustrates the problem. Recall the formal context in Figure 1. From our knowledge, regarding the formal context we already know that:

1. Every odd number is not even, and every even number is not odd.
2. Every number which is greater than 5 is also greater than 2.

Recall also the three attribute implications holding in the formal context. If we consider the second knowledge, the third attribute implication can be inferred from the first attribute implication together with this knowledge.

An attribute implication could be inferred from other attribute implications with backgroud knowledge considered unimportant knowledge or redundant. Therefore, the attribute implication could be ignored. Ignoring an attribute implication will also reduce the size of knowledge extracted from a formal context to obtain only the important knowledge.

Reducing size of knowledge extracted from a formal context is also a recent issue in this research area because the size is sometime very large. The research in [12] reduced the size by congruent relations whereas in [13] by block relations. A research in [14] summarized this issue and classified all recent techniques in reducing the size of knowledge of concept lattice into 3: redundant information removal, simplification, and selection.

Our research could be considered as another technique in redundant information removal. The redundant information means attribute implications which could be inferred from other atribute implications using background knowledge.

Some recent researches in knowledge discovery and data mining had considered background knowledge to ignore or eliminate extracted knowledge which could be inferred using the background knowledge [15, 16, 17, 18, 19, 20]. The inferred exctracted knowledge is also called redundant knowledge. The redundant knowledge have to be eliminated since it becomes a handicap and harder for using it in decision making [15, 17, 20, 21].

Regarding to a formal context, the background knowledge relating with formal context exists. A kind of the background knowledge exists in analysis of many-valued context. As stated earlier, a many-value context has to be transformed into a derived context before being analyzed. The transformation process is called scaling. The scaling needs some scales which are one-valued contexts. A scale can be considered as interpretation of attribute values in the many-valued context. Thus, the scales are representations of prior knowledge to the interpretation. Therefore the scales contain some information which can be seen as background knowledge. Interestingly, many sets of data are in the form of many-valued context [8, 9, 10, 11, 22, 23, 24, 25, 26, 27, 28, 29].

Another kind of background knowledge is from our prior knowledge. The kind of background knowledge exists and some researches used it for formal concept analysis [4, 12, 13, 30, 31, 32, 33]. Some of the researches used such background knowledge to remove or reject some extracted knowledges which are incompatible with it [4, 30, 31, 32, 33] where the extracted knowledge is in the form of attribute implications [4, 30] and concepts [31, 32, 33]. The other researches used such background knowledge to reduce the size of extracted knowledge in the form of concept lattice [12, 13].

To know whether an attribute implication of implicational base can be inferred from some other attribute implications using some background knowledge is a hard problem. However, it probably can be solved using SAT approach. The problem will be encoded into SAT Problem and solved by SAT Solver.

SAT Problem (satisfiability problem) is to determine whether a given propositional formula is satisfiable or not. If it is not, we say that the propositional formula is unsatisfiable. A propositional formula is satisfiable if there is an assignment for all propositonal variables in that formula where the assignment makes the evaluation of the formula to true value. If there is no such assignment, the formula is unsatisfiable [34, 35, 36].

Some algorithms have been developed to solve the SAT Problem and implemented in SAT Solver software. The algorithm which is implemented in many modern SAT Solvers is DPLL algorithm [37, 38, 39]. The DPLL algorithm is a backtracking-based algorithm for deciding the satisfiability of propositional formula in conjunctive normal form. It was introduced in 1962 by Martin Davis, Hilary Putnam, George Logemann and Donald W. Loveland [38] and is a refinement of the earlier Davis-Putnam algorithm, which is a resolution-based procedure developed by Davis and Putnam in 1960 [37].

The recent SAT Solvers are able to solve a propositional formula in millions number of both clauses and variables in reasonable time. It gives a chance to make SAT applicable in real world. Therefore, the current researches in the SAT area are not only focusing in the algorithm [40, 41] and solver [42, 43, 44, 45, 46] but also application of SAT [47].

This paper introduces non-redundant implicational base using scales as background knowledge in many-valued context, models the problem, and formalizes it in the satisfiability problem.

2 FOUNDATIONS

2.1 Formal Context

Definition 1 (Formal Context)**.** A **formal context** (G, M, I) consists of two non-empty sets G and M, and a relation $I \subseteq G \times M$. We call the set G a set of objects, whereas the set M a set of attributes. For $g \in G$ and $m \in M$, $(g, m) \in I$ or gIm is read as the object g has the attribute m [1].

A cross table can represent a formal context. The rows of the cross table represent the objects, and the columns represent the attributes. The headers of the rows are object names, whereas the headers of the columns are attribute names. If an object g has an attribute m, then we cross the table in row g and column m. Figure 1 is a formal context in the cross table.

Definition 2 (Derivation Operator)**.** If $A \subseteq G$ is a set of objects, then we define [1]:

$$A^I = \{m \mid (g, m) \in I \text{ for all } g \in A\}. \tag{1}$$

Reversely, if $B \subseteq M$ is a set of attributes, then we define:

$$B^I = \{g \mid (g, m) \in I \text{ for all } m \in B\}. \tag{2}$$

Notation A^{II} refers to $(A^I)^I$.

2.2 Attribute Implication

Let M a set of attributes in (G, M, I). $A \Rightarrow B$ where $A, B \subseteq M$ is an attribute implication over the formal context. The attribute implication holds in the formal context if each object of the formal context respects the attribute implication. An object $g \in G$ respects the attribute implication iff its attributes set is a model of the implication [2].

Definition 3 (Model of Attribute Implication)**.** Let $A, B, T \subseteq M$. T is a **model of attribute implication** $A \Rightarrow B$ iff $A \not\subseteq T$ or $B \subseteq T$ [2].

Definition 4 (Respecting Object)**.** An object $g \in G$ **respects to** $A \Rightarrow B$ over (G, M, I) iff $\{g\}^I$ is a model of the attribute implication. An object $g \in G$ **respects to** a set $\mathcal{L}$ of attribute implications iff g respects all attribute implications in $\mathcal{L}$ [2].

Definition 5 (Holding Attribute Implication)**.** An attribute implication $A \Rightarrow B$ holds in a formal context (G, M, I) iff all $g \in G$ respect the attribute implication.

```
Algorithm: Implicational Base
Input : A formal context (G,M,I)
Output: The implicational base, 𝓛
begin
  X ← ∅
  𝓛 ← ∅
  repeat
    if (X ≠ X^II) then
      𝓛 ← 𝓛 ∪ {X ⇒ X^II/X}
    X ← Next_Closure(X) from 𝓛
  until (X = M)
  return 𝓛
end
```

Figure 2. Implicational Base algorithm [1, 2]

A set $\mathcal{L}$ of attribute implications holds in a formal context (G, M, I) iff all attribute implications in $\mathcal{L}$ holds in (G, M, I) [2].

Definition 6 (Inference)**.** An implication $A \Rightarrow B$ can be **inferred** from $\mathcal{L}$, denoted by [2]

$$\mathcal{L} \models A \Rightarrow B \tag{3}$$

iff all models of $\mathcal{L}$ are also models of $A \Rightarrow B$.

Definition 7 (Implicational Base)**.** A set $\mathcal{L}$ of attribute implications is an **implicational base** of a formal context, if the followings hold [2]:

- **Sound**, if $\mathcal{L}$ holds in the formal context.
- **Complete**, if the following holds. If there is an attribute implication which holds in the formal context, it can be inferred from $\mathcal{L}$.
- **Non-redundant**, if there is no attribute implication in $\mathcal{L}$ that can be inferred from the others.

Figure 2 shows an algorithm to generate an implicational base of a formal context. **Next_Closure(X) from $\mathcal{L}$** is the lexically smallest model of $\mathcal{L}$ which is lexically larger than X. Let $A, B \subseteq M = \{m_1, m_2, \ldots, m_n\}$ and $m_1 < m_2 < \cdots < m_n$. We define $A < B$, which means "A smaller than B" or "B larger than A", iff $A <_i B$, which is defined as follows, there is i such that

- $i \notin A$ and $i \in B$, and
- for all $j < i$, $j \in A$ iff $j \in B$.

Example 1. Recall a formal context in Figure 1. The implicational base of the formal context generated by algorithm in Figure 2 contains the following attribute implications:

- $\{greater\ than\ 5\} \Rightarrow \{greater\ than\ 2\}$,
- $\{greater\ than\ 2, prime\} \Rightarrow \{odd\}$,
- $\{greater\ than\ 2, greater\ than\ 5, square\} \Rightarrow \{odd\}$,
- $\{odd, prime\} \Rightarrow \{greater\ than\ 2\}$,
- $\{odd, even\} \Rightarrow \{greaterthan\ 2, greaterthan\ 5, prime, square\}$.

2.3 Attribute Implication of Many-Valued Context

Definition 8 (Many-valued Context)**.** A **many-valued context** (G, M, W, I) consists of a set of objects G, a set of attributes M, a set of attribute values W, and a ternary relation $I \subseteq G \times M \times W$ where $(g, m, w) \in I$ and $(g, m, v) \in I$ imply $w = v$ [2, 3].

In the attribute exploration of a many-valued context, we have to transform the many-valued context into one-valued context. The transformation is called scaling. In the scaling, we need some scales, which are also formal contexts [2].

Definition 9 (Scale)**.** A **scale** for attribute $m \in M$ of a many-valued context (G, M, W, I) is a one-valued context $S_m = (G_m, M_m, I_m)$ with $\{w \mid (g, m, w) \in I \text{ and } g \in G\} \subseteq G_m$ [2].

	Final	Written	Practical
1	Pass	Pass	Pass
2	Fail	Pass	Fail
3	Fail	Fail	Pass
4	Fail	Fail	Fail

Figure 3. Many-valued context

	Final:Pass	Final:Fail
Pass	×	
Fail		×

	Written:Pass	Written:Fail
Pass	×	
Fail		×

	Practical:Pass	Practical:Fail
Pass	×	
Fail		×

Figure 4. Scales for attributes: Final, Written, and Practical, respectively

Definition 10 (Derived Context)**.** The **derived context** in scaling of the many-valued context (G, M, W, I) and scales S_m for all $m \in M$ is the context (G, N, J) where

$$N = \bigcup_{m \in M} M_m \tag{4}$$

and for $g \in G$ and $n \in N$, $(g, n) \in J$ iff $(m, g, w) \in I$ and $(w, n) \in I_m$ [2].

Example 2. Figure 3 is an example of a many-valued context with

$$M = \{Final, Written, Practical\}.$$

The many-valued context shows all possible results of driving test. The driving test consists of two parts which are written and practical part showed by attribute *Written* and *Practical*, respectively. The final result which depends on both test parts is showed by attribute *Final*.

By scaling with a formal context in Figure 4 for all attributes in M, we obtain a derived context in Figure 5.

	Final:Pass	Final:Fail	Written:Pass	Written:Fail	Practical:Pass	Practical:Fail
1	×		×		×	
2		×	×			×
3		×		×	×	
4		×		×		×

Figure 5. The derived context

2.4 SAT Problem

We take some notations from [36, 45] and [48] to formulate the propositional formula and the SAT problem.

A propositional formula is a logical formula based on proposition. An atomic (simple) formula consists of a single propositional variable whereas a complex formula is a composition of connectors and propositional variable(s). The connectors are $\wedge$ (conjunction), $\vee$ (disjunction), $\rightarrow$ (implication), $\leftrightarrow$, (biimplication), and $\neg$ (negation).

Definition 11 (Propositional Formula)**.** A **propositional formula** F is recursively defined as follows:

$$F = \begin{cases} p, \\ \neg F', \\ F_1 \circ F_2, \quad \text{where } \circ \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}, \end{cases}$$

where

- p is a propositional variable, possibly with indices,
- F_1, F_2, and F' are propositional formulas.

Definition 12 (Interpretation)**.** An **interpretation** *Int* is a mapping of propositional formulas to truth values $\{\top, \bot\}$.

An interpretation *Int* will uniquely act on each variable occurring in F. Let p a propositional variable. *Int* will be either $Int(p) = \top$ or $Int(p) = \bot$. An interpretation *Int* will be a model of formula F if and only if $Int(F) = \top$. F is satisfiable if and only if F has some models, and F is unsatisfiable if and only if F has no models.

Given a propositional formula F, the goal of the SAT Problem is to determine whether the formula F is satisfiable or unsatisfiable.

3 BACKGROUND KNOWLEDGE IN MANY-VALUED CONTEXT

3.1 Background-Inferring Problem

Given an attribute implication which holds in a derived context, the question is whether the attribute implication can be implied by the other attribute implications, which also hold in the derived context, together with information in its scales.

Definition 13 (Background-inferring Problem)**.** Scales can be considered as interpretations of values in a many-valued context. Those are already some existing knowledges which are used to derive the many-valued context to obtained a derived one-valued context. The implicational base algorithm in Figure 2 does not considered the existing knowledge. The following shows that an attribute implication probably can be inferred from some others attribute implications in the impliational base together with the knowledge in those scales.

Let $\mathcal{L}$ a set of attributes implications which hold in the derived context from a many-valued context (G, M, W, I) and scales S_m for all $m \in M$, $\mathcal{H}$ knowledge represents the scales, and $A \Rightarrow B$ an attribute implication which also holds in the derived context. The **background-inferring problem** is whether:

$$(\mathcal{L} \cup \mathcal{H}) \text{ implies } A \Rightarrow B. \tag{5}$$

It means that all models of $\mathcal{L}$ and $\mathcal{H}$ are also models of $A \Rightarrow B$. Since a scale $S_m = (G_m, M_m, I_m)$ consists of all possible combination values of attributes in M_m, a model T of $\mathcal{L}$ is also a model of $\mathcal{H}$ iff for each S_m, T is compatible with S_m. T is compatible with S_m iff there is $g \in G_m$ such that $\{g\}^{I_m} \subseteq T$ [30].

Example 3. These attribute implications hold in the derived context showed in Figure 5:

- {Practical:Fail} $\Rightarrow$ {Final:Fail},
- {Written:Fail} $\Rightarrow$ {Final:Fail},
- {Written:Pass, Practical:Pass} $\Rightarrow$ {Final:Pass},
- {Final:Fail, Practical:Pass} $\Rightarrow$ {Written:Fail}.

Let $\mathcal{L}$ consist of the three first-attribute-implications and $\mathcal{H}$ represent information from scales in Figure 4. All models of $\mathcal{L}$ containing {Final:Fail, Practical:Pass} are

- {Final:Fail, Final:Pass, Practical:Pass, Written:Pass}, and
- {Final:Fail, Practical:Pass, Written:Fail}.

Because of the scale of attribute Practical (Figure 4), the first model is not the model of $\mathcal{H}$. Thus, only the second model is the model of $(\mathcal{L} \cup \mathcal{H})$. It is also a model of

- {Final:Fail, Practical:Pass} $\Rightarrow$ {Written:Fail}.

Therefore, $(\mathcal{L} \cup \mathcal{H})$ implies the attribute implication.

For the next examples, we will use the natural numbers 1, 2, ... to refer attribute names Final:Pass, Final:Fail, ..., respectively.

4 BACKGROUND-INFERRING PROBLEM IN SAT

The followings are some corresponding notations between formal context and propositional formula in this encoding:

- An attribute $m \in M$ corresponds to a propositional variable p_m.
- $T \subseteq M$ corresponds to an interpretation Int_T. $m \in T$ iff $Int_T(p_m) = \top$.

Proposition 1. Let $T, A, B \subseteq M$. T is a model of $A \Rightarrow B$ iff

$$Int_T\left(\bigwedge_{b \in B}\left(\left(\bigwedge_{a \in A} p_a\right) \rightarrow p_b\right)\right) = \top.$$

Proof.

1. T is a model of $A \Rightarrow B$. There are two possibilities:

   (a) $A \nsubseteq T$
   $\hookrightarrow$ there is $c \in A$, but $c \notin T$
   $\hookrightarrow$ $Int_T(p_c) = \bot$
   $\hookrightarrow$ $Int_T\left(\bigwedge_{a\in A} p_a\right) = \bot$
   $\hookrightarrow$ For all $b \in B$, $Int_T\left(\left(\bigwedge_{a\in A} p_a\right) \to p_b\right) = \top$
   $\hookrightarrow$ $Int_T\left(\bigwedge_{b\in B}\left(\left(\bigwedge_{a\in A} p_a\right) \to p_b\right)\right) = \top$

   (b) $B \subseteq T$
   $\hookrightarrow$ For all $b \in B$, $Int_T(p_b) = \top$
   $\hookrightarrow$ For all $b \in B$, $Int_T\left(\left(\bigwedge_{a\in A} p_a\right) \to p_b\right) = \top$
   $\hookrightarrow$ $Int_T\left(\bigwedge_{b\in B}\left(\left(\bigwedge_{a\in A} p_a\right) \to p_b\right)\right) = \top$

2. $Int_T\left(\bigwedge_{b\in B}\left(\left(\bigwedge_{a\in A} p_a\right) \to p_b\right)\right) = \top$
   $\hookrightarrow$ For all $b \in B$, $Int_T\left(\left(\bigwedge_{a\in A} p_a\right) \to p_b\right) = \top$
   $\hookrightarrow$ There are also two possibilities:

   (a) For all $b \in B$, $Int_T(p_b) = \top$
   $\hookrightarrow$ $B \subseteq T$
   $\hookrightarrow$ T is a model of $A \Rightarrow B$

   (b) $Int_T\left(\bigwedge_{a\in A} p_a\right) = \bot$
   $\hookrightarrow$ There is $c \in A$, such that $Int_T(p_c) = \bot$
   $\hookrightarrow$ There is $c \in A$, but $c \notin T$
   $\hookrightarrow$ $A \nsubseteq T$
   $\hookrightarrow$ T is a model of $A \Rightarrow B$.

□

From Proposition 1, $A \Rightarrow B$ corresponds to a propositional formula:

$$\bigwedge_{b\in B}\left(\left(\bigwedge_{a\in A} p_a\right) \to p_b\right).$$

We will use $F_{A\Rightarrow B}$ to refer the formula.

Example 4. Recall Example 3. We have the following correspond formulas, respectively:

- $p_6 \to p_2$,
- $p_4 \to p_2$,
- $(p_3 \wedge p_5) \to p_1$,
- $(p_2 \wedge p_5) \to p_4$.

Proposition 2. Let $S_m = (G_m, M_m, I_m)$ a scale to obtain a derived context (G, N, J) and $T \subseteq N$. T is compatible with S_m iff

$$Int_T\left(\bigvee_{g\in G_m}\left(\bigwedge_{a\in\{g\}^{I_m}} p_a \wedge \bigwedge_{a\in M_m/\{g\}^{I_m}} \neg p_a\right)\right) = \top.$$

Proof.

1. T is compatible with $S_m = (G_m, M_m, I_m)$
   $\hookrightarrow$ There is $g_c \in G_m$, such that $\{g_c\}^{I_m} \subseteq T$
   $\hookrightarrow$ $Int_T(\bigwedge_{a\in\{g_c\}^{I_m}} p_a \wedge \bigwedge_{a\in M_m/\{g_c\}^{I_m}} \neg p_a) = \top$
   $\hookrightarrow$ $Int_T\left(\bigvee_{g\in G_m}(\bigwedge_{a\in\{g\}^{I_m}} p_a \wedge \bigwedge_{a\in M_m/\{g\}^{I_m}} \neg p_a)\right) = \top$
2. $Int_T\left(\bigvee_{g\in G_m}(\bigwedge_{a\in\{g\}^{I_m}} p_a \wedge \bigwedge_{a\in M_m/\{g\}^{I_m}} \neg p_a)\right) = \top$
   $\hookrightarrow$ There is $g_c \in G_m$, such that $Int_T(\bigwedge_{a\in\{g_c\}^{I_m}} p_a \wedge \bigwedge_{a\in M_m/\{g_c\}^{I_m}} \neg p_a) = \top$
   $\hookrightarrow$ $\{g_c\}^{I_m} \subseteq T$
   $\hookrightarrow$ T is compatible with $S_m = (G_m, M_m, I_m)$.

□

From Proposition 2, we know that the information related with a scale $S_m = (G_m, M_m, I_m)$ corresponds to a propositional formula:

$$\bigvee_{g\in G_m}\left(\bigwedge_{a\in\{g\}^{I_m}} p_a \wedge \bigwedge_{a\in M_m/\{g\}^{I_m}} \neg p_a\right).$$

We will use H_m to refer the propositional formula which a scale S_m corresponds to.

Example 5. Recall Example 3. From scale of attribute Final, Written, and Practical in Figure 4, we have the following formulas:

- $(p_1 \wedge \neg p_2) \vee (\neg p_1 \wedge p_2)$,
- $(p_3 \wedge \neg p_4) \vee (\neg p_3 \wedge p_4)$,
- $(p_5 \wedge \neg p_6) \vee (\neg p_5 \wedge p_6)$.

Proposition 3. T is a model of a set of attribute implications $\mathcal{L}$, iff

$$Int_T\left(\bigwedge_{A\Rightarrow B\in\mathcal{L}} F_{A\Rightarrow B}\right) = \top. \tag{6}$$

Proof. T is a model of $\mathcal{L}$
iff For all $A \Rightarrow B \in \mathcal{L}$, T is also a model of $A \Rightarrow B$
iff For all $A \Rightarrow B \in \mathcal{L}$, $Int_T(F_{A\Rightarrow B}) = \top$ {from Proposition 1}
iff $Int_T\left(\bigwedge_{A\Rightarrow B\in\mathcal{L}} F_{A\Rightarrow B}\right) = \top$. □

Proposition 4. T is a model of $\mathcal{H}$, which is information representing scales $S_m = (G_m, M_m, I_m)$ for all $m \in M$, iff

$$Int_T\left(\bigwedge_{m\in M} H_m\right) = \top. \tag{7}$$

Proof. T is a model of $\mathcal{H}$
iff For all $m \in M$, T is compatible with $S_m = (G_m, M_m, I_m)$
iff For all $m \in M$, $Int_T(H_m) = \top$ {from Proposition 2}
iff $Int_T\left(\bigwedge_{m\in M} H_m\right) = \top$. □

Let $F_{\mathcal{L}} = \bigwedge_{A\Rightarrow B\in\mathcal{L}} F_{A\Rightarrow B}$ and $F_{\mathcal{H}} = \bigwedge_{m\in M} H_m$. $\mathcal{L}$ corresponds to $F_{\mathcal{L}}$, whereas $\mathcal{H}$ corresponds to $F_{\mathcal{H}}$.

Proposition 5. T is a model of $(\mathcal{L} \cup \mathcal{H})$ iff $Int_T(F_{\mathcal{L}} \wedge F_{\mathcal{H}}) = \top$.

Proof. T is a model of $(\mathcal{L} \cup \mathcal{H})$
iff T is a model of both $\mathcal{L}$ and $\mathcal{H}$
iff $Int_T(F_{\mathcal{L}}) = \top$ and $Int_T(F_{\mathcal{H}}) = \top$ {from Proposition 3 and Proposition 4}
iff $Int_T(F_{\mathcal{L}} \wedge F_{\mathcal{H}}) = \top$. □

Proposition 6. $(\mathcal{L}\cup\mathcal{H})$ does not imply $A \Rightarrow B$, iff $F_{\mathcal{L}} \wedge F_{\mathcal{H}} \wedge \neg F_{A\Rightarrow B}$ is satisfiable.

Proof. $(\mathcal{L} \cup \mathcal{H})$ does not imply $A \Rightarrow B$
iff There is $T \in M$, T is a model of $(\mathcal{L} \cup \mathcal{H})$, but T is not a model of $A \Rightarrow B$
iff There is $T \in M$, $Int_T(F_{\mathcal{L}} \wedge F_{\mathcal{H}}) = \top$ (Proposition 5) and $Int_T(F_{A\Rightarrow B}) = \bot$ (Proposition 1)
iff There is $T \in M$, $Int_T(F_{\mathcal{L}} \wedge F_{\mathcal{H}} \wedge \neg F_{A\Rightarrow B}) = \top$
iff $F_{\mathcal{L}} \wedge F_{\mathcal{H}} \wedge \neg F_{A\Rightarrow B}$ is satisfiable. □

Example 6. Recall Example 3, 4, and 5. Let $\mathcal{L} = \{\{6\} \Rightarrow \{2\}, \{4\} \Rightarrow \{2\}, \{3, 5\} \Rightarrow \{1\}\}$ and $\mathcal{H}$ information from scales in Figure 4. We want to check whether $(\mathcal{L}\cup\mathcal{H})$ does not imply $\{2, 5\} \Rightarrow \{4\}$. Then, we obtain the following propositional formula:

1. $p_6 \rightarrow p_2$,
2. $\wedge p_4 \rightarrow p_2$,
3. $\wedge(p_3 \wedge p_5) \rightarrow p_1$,
4. $\wedge((p_1 \wedge \neg p_2) \vee (\neg p_1 \wedge p_2))$,
5. $\wedge((p_3 \wedge \neg p_4) \vee (\neg p_3 \wedge p_4))$,
6. $\wedge((p_5 \wedge \neg p_6) \vee (\neg p_5 \wedge p_6))$,
7. $\wedge\neg((p_2 \wedge p_5) \rightarrow p_4)$.

Let F be the propositional formula. If we consider the conjunct 4 then we only have two possible interpretations e.g. Int_{T_1} and Int_{T_2}, where:

- $Int_{T_1}(p_1) = \top$ and $Int_{T_1}(p_2) = \bot$,
- $Int_{T_2}(p_1) = \bot$ and $Int_{T_2}(p_2) = \top$.

$Int_{T_1}(F) = \bot$ since $Int_{T_1}(\neg((p_2 \wedge p_5) \rightarrow p_4)) = \bot$ (conjunct 7).

Whereas Int_{T_2} will be a model of F, if $Int_{T_2}(p_3) = \bot$ or $Int_{T_2}(p_5) = \bot$ because of conjunct 3. Suppose $Int_{T_2}(p_3) = \bot$. Because of conjunct 5, $Int_{T_2}(p_4) = \top$. It makes Int_{T_2} over conjunct 7 be $\bot$. Thus, $Int_{T_2}(F) = \bot$.

Also Int_{T_2} over conjuct 7 will be $\bot$ if $Int_{T_2}(p_5) = \bot$.

We can conclude that neither Int_{T_1} nor Int_{T_2} will be a model of F. Thus, F is unsatisfiable. Therefore, $(\mathcal{L} \cup \mathcal{H})$ implies $\{2, 5\} \Rightarrow \{4\}$. It is the same conclusion obtained in Example 3.

5 CONCLUSION

We showed that some attribute implications in an implicational base of derived context of many-valued context can be inferred from the many-valued context's scales. Even though, the scales are intepretation of some values in the many-valued context, therefore the scales are an existing knowledge. Some literatures proposed that knowledge in knowledge discovery from data, an implicational base in case of a formal context, which can be inferred from existing or background knowledge should be eliminated. They will be redundant knowledge.

We also formalized a model to check the redundancy in SAT Problem. The formulation has also been proven.

In the next research we will develop an algorithm to obtain non-redundant implicational base of many-valued context using scales as background knowledge based on the proposed model. Some experiments with real data also will be conducted using the algorithm.

REFERENCES

[1] Ganter, B.—Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer Verlag, Berlin, Germany, 1999, doi: 10.1007/978-3-642-59830-2.

[2] Ganter, B.: Formal Concept Analysis: Algorithmic Aspects. Available at: `http://www.math.tu-dresden.de/~ganter/cl03/cl02.pdf`, 2002.

[3] Gugisch, R.: Many-Valued Context Analysis Using Descriptions. In: Delugach, H. S., Stumme, G. (Eds.): Conceptual Structures: Broadening the Base (ICCS 2001). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 2120, 2001, pp. 157–168, doi: 10.1007/3-540-44583-8_12.

[4] Bělohlávek, R.—Vychodil, V.: Formal Concept Analysis with Background Knowledge: Attribute Priorities. IEEE Transactions on Systems, Man, and Cybernetics, Part C, Vol. 39, 2009, No. 4, pp. 399–409, doi: 10.1109/TSMCC.2008.2012168.

[5] Xu, Y.—Wang, K.—Zhang, B.—Chen, Z.: Privacy-Enhancing Personalized Web Search. Proceedings of the 16th International Conference on World

Wide Web (WWW '07), Alberta, Canada, ACM, 2007, pp. 591–600, doi: 10.1145/1242572.1242652.

[6] Kumar, C. A.: Knowledge Discovery in Data Using Formal Concept Analysis and Random Projections. International Journal of Applied Mathematics and Computer Science, Vol. 21, 2011, No. 4, pp. 745–756, doi: 10.2478/v10006-011-0059-1.

[7] Klimeš, J.: Using Formal Concept Analysis for Control in Cyber-Physical Systems. Procedia Engineering, Vol. 69, 2014, pp. 1518–1522, doi: 10.1016/j.proeng.2014.03.149.

[8] Kaytoue, M.—Kuznetsov, S. O.—Napoli, A.—Duplessis, S.: Mining Gene Expression Data with Pattern Structures in Formal Concept Analysis. Information Sciences, Vol. 181, 2011, No. 10, pp. 1989–2001, doi: 10.1016/j.ins.2010.07.007.

[9] Du, Y.—Li, H.: Strategy for Mining Association Rules for Web Pages Based on Formal Concept Analysis. Applied Soft Computing, Vol. 10, 2010, No. 3, pp. 772–783, doi: 10.1016/j.asoc.2009.09.007.

[10] Lee, C.—Jeon, J.—Park, Y.: Monitoring Trends of Technological Changes Based on the Dynamic Patent Lattice: A Modified Formal Concept Analysis Approach. Technological Forecasting and Social Change, Vol. 78, 2011, No. 4, pp. 690–702, doi: 10.1016/J.TECHFORE.2010.11.010.

[11] Beydoun, G.: Formal Concept Analysis for an e-Learning Semantic Web. Expert Systems with Applications, Vol. 36, 2009, No. 8, pp. 10 952–10 961, doi: 10.1016/j.eswa.2009.02.023.

[12] Viaud, J.-F.—Bertet, K.—Missaoui, R.—Demko, C.: Using Congruence Relations to Extract Knowledge from Concept Lattices. Discrete Applied Mathematics, Vol. 249, 2018, pp. 135–150, doi: 10.1016/j.dam.2016.11.021.

[13] Konecny, J.—Krupka, M.: Block Relations in Formal Fuzzy Concept Analysis. International Journal of Approximate Reasoning, Vol. 73, 2016, pp. 27–55, doi: 10.1016/j.ijar.2016.02.004.

[14] Dias, S. M.—Vieira, N. J.: Concept Lattices Reduction: Definition, Analysis and Classification. Expert Systems with Applications, Vol. 42, 2015, No. 20, pp. 7084–7097, doi: 10.1016/j.eswa.2015.04.044.

[15] Diaz, J.—Molina, C.—Vila, M. A.: A Model for Redundancy Reduction in Multidimensional Association Rules. In: dos Reis, A. P., Abraham, A.P. (Eds.): Proceedings of IADIS European Conference Data Mining 2013 (part of MCCSIS 2013). IADIS Press, Lisbon, Portugal, 2013, pp. 89–93.

[16] Nguyen, D.—Nguyen, L. T. T.—Vo, B.—Hong, T.-P.: A Novel Method for Constrained Class Association Rule Mining. Information Sciences, Vol. 320, 2015, pp. 107–125, doi: 10.1016/j.ins.2015.05.006.

[17] Ashrafi, M. Z.—Taniar, D.—Smith-Miles, K.: A New Approach of Eliminating Redundant Association Rules. In: Galindo, F., Takizawa, M., Traunmüller, R. (Eds.): Database and Expert Systems Applications (DEXA 2004). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 3180, 2004, pp. 465–474, doi: 10.1007/978-3-540-30075-5_45.

[18] Baralis, E.—Cagliero, L.—Cerquitelli, T.—Garza, P.: Generalized Association Rule Mining with Constraints. Information Sciences, Vol. 194, 2012, pp. 68–84, doi: 10.1016/j.ins.2011.05.016.

[19] Chandanan, A. K.—Shukla, M. K.: Removal of Duplicate Rules for Association Rule Mining from Multilevel Dataset. Procedia Computer Science, Vol. 45, 2015, pp. 143–149, doi: 10.1016/j.procs.2015.03.106.

[20] Diaz, J.—Molina, C.—Vila, M. A.: Using Imprecise User Knowledge to Reduce Redundancy in Association Rules. Proceedings of the 2015 Conference of the International Fuzzy Systems Association and the European Society for Fuzzy Logic and Technology, Advances in Intelligent Systems Research Series, 2015, pp. 1098–1105, doi: 10.2991/ifsa-eusflat-15.2015.155.

[21] Gondek, D.—Hofmann, T.: Non-Redundant Data Clustering. Knowledge and Information Systems, Vol. 12, 2007, No. 1, pp. 1–24, doi: 10.1007/s10115-006-0009-7.

[22] Muangprathub, J.—Boonjing, V.—Pattaraintakorn, P.: A New Case-Based Classification Using Incremental Concept Lattice Knowledge. Data and Knowledge Engineering, Vol. 83, 2013, pp. 39–53, doi: 10.1016/j.datak.2012.10.001.

[23] Kaytoue-Uberall, M.—Duplessis, S.—Napoli, A.: Using Formal Concept Analysis for the Extraction of Groups of Co-Expressed Genes. In: Le Thi, H. A., Bouvry, P., Pham Dinh, T. (Eds.): Modelling, Computation and Optimization in Information Systems and Management Sciences (MCO 2008). Springer, Berlin, Heidelberg, Communications in Computer and Information Science, Vol. 14, 2008, pp. 439–449, doi: 10.1007/978-3-540-87477-5_47.

[24] Castellanos, A.—Cigarrán, J.—García-Serrano, A.: Formal Concept Analysis for Topic Detection: A Clustering Quality Experimental Analysis. Information Systems, Vol. 66, 2017, pp. 24–42, doi: 10.1016/j.is.2017.01.008.

[25] Cigarrán, J.—Castellanos, Á.—García-Serrano, A.: A Step Forward for Topic Detection in Twitter: An FCA-Based Approach. Expert Systems with Applications, Vol. 57, 2016, pp. 21–36, doi: 10.1016/j.eswa.2016.03.011.

[26] Fu, G.: FCA Based Ontology Development for Data Integration. Information Processing and Management, Vol. 52, No. 5, 2016, pp. 765–782, doi: 10.1016/j.ipm.2016.02.003.

[27] Valverde-Albacete, F. J.—González-Calabozo, J. M.—Peñas, A.—Peláez-Moreno, C.: Supporting Scientific Knowledge Discovery with Extended, Generalized Formal Concept Analysis. Expert Systems with Applications, Vol. 44, 2016, pp. 198–216, doi: 10.1016/j.eswa.2015.09.022.

[28] Lihonosova, A.—Kaminskaya, A.: Using Formal Concept Analysis for Finding the Closest Relatives Among a Group of Organisms. Procedia Computer Science, Vol. 31, 2014, pp. 860–868, doi: 10.1016/j.procs.2014.05.337.

[29] Keller, B. J.—Eichinger, F.—Kretzler, M.: Formal Concept Analysis of Disease Similarity. Proceedings of the 2012 AMIA Joint Summits on Translational Science, 2012, pp. 42–51.

[30] Ganter, B.: Attribute Exploration with Background Knowledge. Theoretical Computer Science, Vol. 217, 1996, No. 2, pp. 215–233, doi: 10.1016/S0304-3975(98)00271-0.

[31] Bělohlávek, R.—Sklenář, V.—Zacpal, J.: Concept Lattices Constrained by Attribute Dependencies. In: Pokorný, J., Richta, K. (Eds.): Proceedings of the Dateso 2004 Annual International Workshop on Databases, Texts, Specifications and Objects, Desna, Czech Republic, 2004. CEUR Workshop Proceedings, Vol. 98, 2004, pp. 63–73.

[32] Bělohlávek, R.—Sklenář, V.: Formal Concept Analysis Constrained by Attribute-Dependency Formulas. In: Ganter, B., Godin, R. (Eds.): Formal Concept Analysis (ICFCA 2005). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 3403, 2005, pp. 176–191, doi: 10.1007/978-3-540-32262-7_12.

[33] Belohlavek, R.—Vychodil, V.: Adding Background Knowledge to Formal Concept Analysis via Attribute Dependency Formulas. Proceedings of the 2008 ACM Symposium on Applied Computing (SAC '08), Fortaleza, Ceara, Brazil, 2008, pp. 938–943, doi: 10.1145/1363686.1363900.

[34] Posthoff, C.—Steinbach, B.: SAT-Problems – New Findings. Proceedings of 9th WSEAS International Conference on Data Networks, Communications, Computers, WSEAS Press, 2007, pp. 339–344.

[35] Posadas-Causor, A.—Torres-Jimenez, J.: SAT-DB: An Integrated System for Satisfiability Problem Study. Proceedings of 6th WSEAS International Multiconference on Circuits, Systems, Communications and Computers, WSEAS Press, 2002, pp. 4951–4956.

[36] Biere, A.—Heule, M.—van Maaren, H.—Walsh, T.: Handbook of Satisfiability. Amsterdam, IOS Press, Frontiers in Artificial Intelligence and Applications, Vol. 185, 2009, 980 pp.

[37] Davis, M.—Putnam, H.: A Computing Procedure for Quantification Theory. Journal of the ACM, Vol. 7, 1960, No. 3, pp. 201–215, doi: 10.1145/321033.321034.

[38] Davis, M.—Logemann, G.—Loveland, D.: A Machine Program for Theorem-Proving. Communications of the ACM, Vol. 5, 1962, No. 7, pp. 394–397, doi: 10.1145/368273.368557.

[39] Nieuwenhuis, R.—Oliveras, A.—Tinelli, C.: Solving SAT and SAT Modulo Theories: From an Abstract Davis–Putnam–Logemann–Loveland Procedure to DPLL(T). Journal of the ACM, Vol. 53, 2006, No. 6, pp. 937–977, doi: 10.1145/1217856.1217859.

[40] Martinez-Rios, F.—Frausto-Solis, J.: An Hybrid Simulated Annealing Threshold Accepting Algorithm for Satisfiability Problems Using Dynamically Cooling Schemes. WSEAS Transactions on Computers, Vol. 7, 2008, No. 5, pp. 374–386.

[41] Vázquez-Morán, I.—Torres-Jiménez, J.: A SAT Instances Construction Based on Hypergraphs. Proceedings of the Fifth International Conference on Computational Intelligence and Multimedia Applications (ICCIMA '03), 2003, pp. 244–247.

[42] Moskewicz, M. W.—Madigan, C. F.—Zhao, Y.—Zhang, L.—Malik, S.: Chaff: Engineering an Efficient SAT Solver. Proceedings of the 38th Annual Design Automation Conference, ACM, 2001, pp. 530–535, doi: 10.1145/378239.379017.

[43] Manthey, N.: Coprocessor – A Standalone SAT Preprocessor. In: Tompits, H. et al. (Eds.): Applications of Declarative Programming and Knowledge Management (INAP 2011, WLP 2011). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 7773, 2013, pp. 297–304, doi: 10.1007/978-3-642-41524-1_18.

[44] Manthey, N.—Saptawijaya, A.: Towards Improving the Resource Usage of SAT-Solvers. In: Le Berre, D. (Ed.): POS-10. Pragmatics of SAT, Edinburgh, UK, 2010, EasyChair, Vol. 8, 2012, pp. 28–40, doi: 10.29007/3vwv.

[45] Hyvärinen, A. E. J.—Manthey, N.: Designing Scalable Parallel SAT Solvers. In: Cimatti, A., Sebastiani, R. (Eds.): Theory and Applications of Satisfiability Testing (SAT 2012). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 7317, 2012, pp. 214–227, doi: 10.1007/978-3-642-31612-8_17.

[46] Manthey, N.: Towards Next Generation Sequential and Parallel SAT Solvers. KI – Künstliche Intelligenz, Vol. 30, 2016, No. 3, pp. 339–342, doi: 10.1007/s13218-015-0406-8.

[47] Lynce, I.—Ouaknine, J.: Sudoku as a SAT Problem. Proceedings of the Ninth International Symposium on Artificial Intelligence and Mathematics (AIMATH 06). Springer, Berlin, 2006.

[48] Huth, M.—Ryan, M.: Logic in Computer Science: Modelling and Reasoning about Systems. 2nd Edition. Cambridge University Press, Cambridge, 2004, doi: 10.1017/cbo9780511810275.

Taufiq Hidayat is Ph.D. candidate at the Fakulti Teknologi Maklumat dan Komunikasi (FTMK), Universiti Teknikal Malaysia Melaka (UTeM), Malaysia. He currently works as Junior Lecturer at Universitas Islam Indonesia, Indonesia. His main research areas include formal concept analysis, mathematical logic, machine learning and data mining.

Asmala bin Ahmad is Associate Professor at the Fakulti Teknologi Maklumat dan Komunikasi (FTMK), Universiti Teknikal Malaysia Melaka (UTeM), Malaysia. He also serves as the research coordinator for the faculty. His research interest includes remote sensing, image processing, artificial intelligence and applied mathematics.

Mohammad Ishak bin Desa is Professor at the Fakulti Teknologi Maklumat dan Komunikasi (FTMK), Universiti Teknikal Malaysia Melaka (UTeM), Malaysia. His main research areas include operations research, computational intelligence, data science and analytics.