USING SAT FOR ATTRIBUTE EXPLORATION OF FORMAL CONTEXT WITH CONSTRAINT

Taufiq Hidayat

Informatics Department, Islamic University of Indonesia
Yogyakarta / Indonesian
taufiq.hidayat@uii.ac.id

Abstract

This paper presents a formulation of formal context satisfying some constraints which we already know, and an encoding a constraint-implying problem into SAT problem. The constraint-implying problem is whether an attribute implication which holds in the formal context is implied by the other attribute implications which also hold in the formal context together with information of the constraints. This problem occurs in attribute exploration of formal context.

Keywords - Attribute exploration, constraint, SAT problem.

1 INTRODUCTION

Recently, studying attribute exploration of formal context considers background knowledge. In this attribute exploration, some attribute implications are ignored if they are implied by some other attribute implications together with the background knowledge [2][6][7]. In [15], the problem to check whether an attribute exploration is implied by some other attribute implications together with background knowledge is called a background-implying problem.

In [2][5], attribute exploration for many-valued context is presented. A problem in attribute exploration for its derived context is similar with a problem for formal context with background knowledge. In this case, all scales of the many-valued context are considered as the background knowledge of the derived context.

Furthermore the background-implying problem in attribute exploration for many-valued context is encoded into SAT (satisfiability) problem in [15], which can be solved by SAT solver. SAT problem is interesting since any problem can be encoded into a SAT problem to solve it[8]. Many studies concerning this area have been done e.g studies in [13] and [14]. Researches about the problem do not only concern in theoretical aspect but also in implementation and application. Some algorithms and some SAT solvers to solve the problem have been developed [9][10][11][12]. Recently, a SAT solver can solve a SAT problem with large clauses and large number of variables in reasonable time. However, many SAT solvers only solve a propositional formula in Conjunction Normal Form (CNF).

A constraint is another form of background knowledge. The constraint restricts attribute-values of a formal context. A formal context satisfies the constraint if and only if each object has attributes which satisfy the constraint. Suppose we already know that a formal context satisfying some constraints. We get the same problem in attribute exploration with background knowledge which is a set of constraints in this case. We will call the problem the constraint-implying problem.

This paper will propose a formulation of a formal context with constraints and an encoding the constraint-implying problem into a SAT problem such that the problem can be solved by SAT solver. However, we leave the problem in general propositional formula without converting the formula into CNF.

2 FOUNDATION

2.1 Formal Context

We rewrite some definition from our previous paper in [15].

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.[3]* □

A cross table can represent a formal context *(G,M,I)*, where rows represent *G* and columns represent *M*. A cell of the table in row *g* and column *m* represents a relation *I* of object $g \in G$ and attribute $m \in M$. We cross the cell if $(g,m) \in I$. Fig. 1 is an example of formal context in a cross table.

Definition 2. Derivation Operator

If $A \subseteq G$ and $B \subseteq M$ is a set of objects, then we define [3]:

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

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

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

Definition 3. 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 $(g,m,v) \in I$ implies $w=v$. [1][4][5].* □

Scaling transforms a many-valued context into a one-valued context by some scales which are also formal contexts. We call the one-valued context the derived context.

Definition 4. Scale

*A **scale** for attribute $m \in M$ of a many-valued context (G,M,W,I) is a one-valued context $S_m = (G_m, M_m, S_m)$ where $\{w \mid (g,m,w) \in I \quad and \quad g \in G\} \subseteq G_m$.* □

Definition 5. 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*

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

and for $g \in G$ and $n \in N$: $(g,n) \in J$ *iff* $(g,m,w) \in I$ *and* $(w,n) \in I_m$ [1]. □

A. Attribute Exploration

An implication in the form $A \Rightarrow B$ where $A,B \subseteq M$ is an attribute implication over a formal context *(G,M,I)*. The attribute implication holds in the formal context iff each object respects it[1].

Definition 6. 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 $B \subseteq T$.* □

Definition 7. Object Respecting

*An object $g \in G$ **respects** $A \Rightarrow B$ over (G,M,I) iff g^I is a model of the attribute implication.* □

B. Attribute Exploration of Many-Valued Context

For attribute exploration of many-valued context, we define the background-implying problem which is whether an attribute implication holding in its derived-context is implied by the other ones holding also in the derived-context together with its scales.

Definition 8. Background-implying Problem[15]

*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 $m \in M$, $\mathcal{H}$ information representing the scales, and $A \Rightarrow B$ an attribute implication which also holds in the derived context. The **background-implying problem** is whether[5]:*

$\mathcal{L} \cup \mathcal{H}$ implies $A \Rightarrow B$. □

It means that all models of $\mathcal{L}$ and $\mathcal{H}$ are also models of $A \Rightarrow B$.[4][15]

2.2 Constraint

A constraint on a set of variables is a restriction on the values that they can take simultaneously. A constraint can be represented in many ways. However, a constraint can be represented as a set which contains all the legal compound labels for the variables[16].

Definition 9. Label

*Let $\mathcal{W}$ a finite set of variables and D_x a domain of $x \in \mathcal{W}$. A **label** in $\mathcal{W}$ is a pair <x,v> where $x \in \mathcal{W}$ and $v \in D_x$, which means that a value v is assigned to a variable x.* □

Definition 10. Compound Label

*Let $<x_i,v_i>$ a label in $\mathcal{W}$. A **compound label** over $\mathcal{W}$ is $L_{\mathcal{W}} = (<x_1,v_1><x_2,v_2>\ldots<x_n,v_n>)$ which means that values $v_1, v_2, \ldots, v_n$ are assigned to variabels $x_1, x_2, \ldots, x_n$, respectively.* □

Definition 11. Constraint

*Let $\mathcal{S} = \{x_1, x_2, \ldots, x_n\}$. A **constraint** on set $\mathcal{S}$, is denoted by $C_{\mathcal{S}}$, is a set of legal compound labels, which each compound label is in the form $(<x_1,v_1><x_2,v_2>\ldots<x_n,v_n>)$.* □

Definition 12. Constraint Satisfying

*Let $\mathcal{S}$ and $\mathcal{W}$ finite sets. A compound label $L_{\mathcal{W}}$ **satisfies** $C_{\mathcal{S}}$ iff there is a compound label $L \in C_{\mathcal{S}}$ such that every pair <x,v> in L is also a pair in $L_{\mathcal{W}}$.* □

Example 1

Let $\mathcal{S} = \{x_1, x_2\}$, $D_{x1}=D_{x2}=\{1,2,3,4\}$ and $C_{\mathcal{S}} = \{(<x_1,1>,<x_2,2>),(<x_1,2>,<x_2,3>),(<x_1,3>,<x_2,4>)\}$. The compound label $(<x_1,2>,<x_2,3>)$ satisfies $C_{\mathcal{S}}$, where as the compound label $(<x_1,2>,<x_2,2>)$ does not satisfy $C_{\mathcal{S}}$. □

2.3 SAT Problem

We take some notations from [8] and [9] to formulate the propositional formula and the SAT problem. Let *p*, *q*, possibly with índices be propositional variables and $\perp$, T be truth values denoting false and true, respectively.

Definition 13. Propositional Formula

Let p, q, possibly with índices be propositional variables. A propositional formulas F is defined as follows:

$$F = \begin{cases} v \\ \neg F_1 \\ F_1 * F_2 \end{cases}$$

where:

- *v* : *a propositional variable, p or q*
- F_1, F_2 : *propositional formulas*
- $\neg$: *negation operation*
- $*$: *either $\vee, \wedge, \rightarrow$, or $\leftrightarrow$ which are disjunction, conjunction, implication, or bi-implication operation respectively.* □

Definition 14. Interpretation

An interpretation Int is a mapping propositional formulas to truth values {T, $\perp$}. □

An interpretation *Int* will uniquely act on each variable occurring in *F*. Let *p* a propositional variable. *Int* will be either $Int(p) = T$ or $Int(p) = \perp$.

Example 2

Let $F = (p_1 \vee p_2) \rightarrow (p_1 \wedge p_2)$. If $Int(p_1)$=T and $Int(p_2)=\perp$ then $Int(F)=\perp$. If $Int(p_1)$=T and $Int(p_2)$=T then $Int(F)$=T. □

Definition 15. Model, Satisfiable, Unsatisfiable

An interpretation Int will be a ***model*** *of formula F iff Int(F) = T. F is* ***satisfiable*** *iff F has some models, and F is* ***unsatisfiable*** *iff F has no models*. □

Example 3

1. $(p_1 \vee p_2) \rightarrow (p_1 \wedge p_2)$ is satisfiable since these interpretations are models of *the formula*:
    a. $Int(p_1)$=T and $Int(p_2)$=T.
    b. $Int(p_1)=\perp$ and $Int(p_2)=\perp$.
2. $(p_1 \wedge p_2) \wedge (\neg p_1 \vee \neg p_2)$ is unsatisfiable since the formula has no models. □

Definition 16. SAT Problem

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

SAT solver is software to solve SAT problem of propositional formula in CNF (Conjunction Normal Form). Thus, we have to convert any propositional formulas into CNF[8].

Definition 17. CNF Formula

A propositional formula F is in CNF if the formula is in the form:

$$F = (l_{1,1} \vee l_{1,2} \vee \ldots \vee l_{1,m1}) \wedge (l_{2,1} \vee l_{2,2} \vee \ldots \vee l_{2,m2}) \wedge \ldots \wedge (l_{n,1} \vee l_{n,2} \vee \ldots \vee l_{n,mn})$$

where $l_{i,j}$ is either p or ¬p for any propositional variable p. □

Example 4

Recall Example 3. Respectively, the propositional formulas in CNF are follows:

1. $(\neg p_1 \vee p_2) \wedge (p_1 \vee \neg p_2)$
2. $p_1 \wedge p_2 \wedge (\neg p_1 \vee \neg p_2)$ □

3 FORMAL CONTEXT WITH CONSTRAINT

3.1 Constraints for a Formal Context

Suppose we have a formal context *(G,M,I)*. We define a variables set $\mathcal{S} = \{x_P \mid P \subseteq M\}$. We also define a domain for each variable x_P is $D_P = 2^P$. Here we want to give a constraint to restrict some attributes of $P \subseteq M$ for each object in *G*. Then, we can define a constraint

$$C_{\{x_P\}} = \{(< x_P, v_P >) \mid v_P \in D_P\} \quad (3)$$

Example 5

Fig. 1 shows a formal context of "Bodies of Water". From our previous knowledge, we know well that there is a constraint for attributes *stagnant-running*. The constraint for the attributes is each object in the formal context having exactly one attribute of both, either *stagnant* or *running*. There are also similar constraints for attributes *inland-maritime* and *constant-temporary*. Let P_1 = {*stagnant, running*}, P_2 = {*inland, maritime*}, and P_3 = {*constant, temporary*}. Then, we have three constraints for the formal context, i.e.:

$$C_{\{x_{P_1}\}} = \{(< x_{P_1}, \{stagnant\}>), (< x_{P_1}, \{running\}>)\}$$

$$C_{\{x_{P_2}\}} = \{(< x_{P_2}, \{inland\}>), (< x_{P_2}, \{maritime\}>)\}$$

$C_{\{x_{P3}\}} = \{(<x_{P_3}, \{constant\}>), (<x_{P_3}, \{temporary\}>)\}$ □

An object $g \in G$ satisfies a constraint $C_{\{x_P\}}$ iff the attributes combination belonging to g in P is a value assigned to x_P in the constraint. For example, object *tarn* satisfies three constraints in Example 5 since attributes combination belonging to g in P_1 is *{stagnant}*, in P_2 is *{inland}*, and in P_3 is *{constant},* which are assigned to x_{P_1} in $C_{\{x_{P1}\}}$, x_{P_2} in $C_{\{x_{P2}\}}$, and x_{P_3} in $C_{\{x_{P3}\}}$, respectively.

Definition 18. Constraint Satisfying Object

An object $g \in G$ of formal context (G,M,I) ***satisfies*** *a constraint $C_{\{x_P\}}$ where $P \subseteq M$* iff a compound label $L_{\{x_P\}} = (<x_P, g^I \cap P>)$ *satisfies the constraint.* □

Definition 19. Constraint Satisfying Formal-Context

A formal context (G,M,I) ***satisfies*** *a constraint $C_{\{x_P\}}$ iff for all $g \in G$, g satisfies the constraint.* □

It is trivial to check that the formal context of "Bodies of Water" satisfies the three constraints in Example 5.

	natural	artificial	stagnant	running	inland	maritime	constant	temporary
tarn	×		×		×		×	
trickle	×			×	×		×	
stream	×			×	×		×	
torrent	×			×	×		×	
river	×			×	×		×	
channel				×	×		×	
canal		×		×	×		×	
lagoon	×		×			×	×	
lake	×		×		×		×	
mere		×	×		×		×	
plash	×		×		×			×
pond		×	×		×		×	
pool	×		×		×		×	
puddle	×		×		×			×
reservoir		×	×		×		×	
sea	×		×			×	×	

Fig. 1. Formal Context of "Bodies of Water"[2]

3.2 Representing a Constraint as a Formal Context

Interestingly, we can represent a constraint as a formal context. Let $C_{\{x_P\}}$ a constraint of formal context *(G,M,I)*. The constraint is able to be represented as a formal context (G_P,M_P,I_P) which is defined as follows:

- $G_P = C_{\{x_P\}}$
- $M_P = P$
- $(g,m) \in I_P$ where $g = (<x_P,A>) \in G_P$ and $m \in M_P$ iff $m \in A$.

Example 6

Recall Example 5. The constraints $C_{\{x_{P_1}\}}$, $C_{\{x_{P_2}\}}$, and $C_{\{x_{P_3}\}}$ are represented respectively as the following formal contexts:

	stagnant	running
$(< x_{P_1}, \{stagnant\}>)$	×	
$(< x_{P_1}, \{running\}>)$		×

	inland	maritime
$(< x_{P_2}, \{inland\}>)$	×	
$(< x_{P_2}, \{maritime\}>)$		×

	constant	temporary
$(< x_{P_3}, \{constant\}>)$	×	
$(< x_{P_3}, \{temporary\}>)$		×

□

The object names in this representation are not important. An important thing we need to note is that each object of the formal context is associated to a label of the represented constraint.

Proposition 1

Let (G_P,M_P,I_P) the representation of constraint $C_{\{x_P\}}$. A formal context (G,M,I) satisfies a constraint (G_P,M_P,I_P) iff for all $g \in G$, there is $g_P \in G_P$ such that $g^I \cap M_P = g_P^{I_P}$.

Proof:

(G,M,I) satisfies (G_P,M_P,I_P) iff *(G,M,I) satisfies* $C_{\{x_P\}}$ **iff** for all $g \in G$, g satisfies $C_{\{x_P\}}$.

$g \in G$ satisfies $C_{\{x_P\}}$ **iff** a compound label $L_{\{x_P\}} = (<x_P, g^I \cap P>)$ *satisfies* $C_{\{x_P\}}$

iff there is a label $(<x_P, A>) \in C_{\{x_P\}}$, such that $g^I \cap P = A$

iff there is $g_P \in G_P$, which associated to the label, such that $g^I \cap P = g_P^{I_P}$ (since $g_P^{I_P} = A$)

iff there is $g_P \in G_P$, such that $g^I \cap M_P = g_P^{I_P}$. □

4 CONSTRAINT-IMPLYING PROBLEM

We will define the constraint-implying problem. For this case, we already know that a formal context satisfies some constraints.

Definition 20. Constraint-Implying Problem

Given an attribute implication $A \Rightarrow B$ which holds in a formal context (G,M,I), a set of attribute implications $\mathcal{L}$ which also hold in the formal context, and n constraints $C_{\{x_{P_1}\}}$, $C_{\{x_{P_2}\}}$, … ,$C_{\{x_{P_n}\}}$ which the formal context satisfies. *The **constraint-implying problem** is whether:*

$\mathcal{L} \cup \mathcal{K}$ implies $A \Rightarrow B$

where $\mathcal{K}$ is a representation of the constraints. □

4.1 Background-Implying Problem is also Constraint-Implying Problem

The difference between the constraint-implying problem and background-implying problem is the information of $\mathcal{K}$ and $\mathcal{H}$. We will prove that $\mathcal{H}$ in the background-implying problem is similar to $\mathcal{K}$ in the constraint-implying problem. For that purpose, it is sufficient to proof that $\mathcal{H}$ is also information of constraints.

Proposition 2

$\mathcal{H}$ in the background-implying problem is also information of constraints which the derived context satisfies.

Proof:

$\mathcal{H}$ in the background-implying problem is information of scales. Thus, we will prove that scales are constraints which its derived-context satisfies.

Let (G,N,J) a derived context of many-valued context (G,M,W,I) and $S_m=(G_m,M_m,I_m)$ a scale. (G,N,J) satisfies the constraint $S_m=(G_m,M_m,I_m)$ **iff** for all $g \in G$, there is $g_m \in G_m$ such that $g^J \cap M_m = g_m^{I_m}$ (Proposition 1)

Let $g \in G$ and $w \in W$ such that $(g,m,w) \in I$. By definition, we know that $w \in G_m$ and for all $n \in M_m \subseteq N$, $(g,n) \in J$ **iff** $(w,n) \in I_m$. Thus, $g^J \cap M_m = w^{I_m}$.

Therefore, for all $g \in G$, there is always $w \in G_m$, where $(g,m,w) \in I$, such that $g^J \cap M_m = w^{I_m}$. Then, (G,N,J) satisfies the constraint $S_m=(G_m,M_m,I_m)$. □

4.2 Encoding Constraint-Implying Problem into SAT Problem

From Proposition 2, we know that each scale is a constraint in $\mathcal{H}$ of background-implying problem. Since each scale which is also a constraint can be encoded independently[15], we can encode a constraint in the same way for the constraint-implying problem although the constraint-implying problem is more general than the background-implying problem.

Thus, from our work in [15],

$\mathcal{L} \cup \mathcal{K}$ does not imply $A \Rightarrow B$

if and only if the following propositional formulas are satisfiable:

- $\bigwedge_{d \in D} \left(\wedge_{c \in C} p_c \to p_d \right)$ for each $C \Rightarrow D \in \mathcal{L}$
- $\left(\bigvee_{g \in G} \left(\left(\wedge_{a \in g'} p_a \right) \wedge \left(\wedge_{a \in M / g'} \neg p_a \right) \right) \right)$ for each constraint (G,M,I) in $\mathcal{K}$
- $\neg \left(\bigwedge_{b \in B} \left(\wedge_{a \in A} p_a \to p_b \right) \right)$

Example 7

Recall Example 5 and Example 6. We use natural number *1,2,…,8* instead of attributes *natural, artificial, stagnant, running, inland, maritime, constant,* and *temporary,* resp. Let

$\mathcal{L}$ = {{8}⇒{1,3,5},{6}⇒{1,3,7}}

$\mathcal{K}$ is information of constraints $C_{\{x_{P1}\}}$, $C_{\{x_{P2}\}}$, and $C_{\{x_{P3}\}}$

$\mathcal{L} \cup \mathcal{K}$ does not imply {4}⇒{5,7}, if only if the following formulas are satisfiable:

$(p_8 \to p_1) \wedge (p_8 \to p_3) \wedge (p_8 \to p_5)$

$(p_6 \to p_1) \wedge (p_6 \to p_3) \wedge (p_6 \to p_7)$

$(p_3 \wedge \neg p_4) \vee (\neg p_3 \wedge p_4)$

$(p_5 \wedge \neg p_6) \vee (\neg p_5 \wedge p_6)$

$(p_7 \wedge \neg p_8) \vee (\neg p_7 \wedge p_8)$

$\neg ((p_4 \to p_5) \wedge (p_4 \to p_7))$ □

5 CONCLUSION

We have proposed a formulation of a formal context with constraints and an encoding of a constraints-implying problem into a SAT problem. The constraint-implying problem occurs in attribute exploration of formal context with background knowledge which is a set of constraints in this case. However, we need a step to convert the propositional formula into CNF before a SAT solver can solve the encoded SAT problem.

6 FUTURE WORKS

The next research is to find a best method for converting the encoded propositional formula into CNF since a bad CNF formula can make performance of a SAT solver worse. Another next research is to develop some applications of the attribute exploration of formal context with constraint in real world.

7 REFERENCES

[1] B. Ganter, and R. Wille, Formal Concept Analysis: Mathematical Foundations, Springer Verlag, Berlin, 1999.

[2] R. Wille, “Formal Concept Analysis as Mathematical Theory of Concepts and Concept Hierarchies”, Formal Concept Analysis: Foundation and Applications, LNCS, vol. 3626. Berlin: Springer, 2005, pp. 1—33.

[3] B. Ganter, and R. Wille, Formal Concept Analysis: Mathematical Foundations, Springer Verlag, Berlin, 1999.

[4] B. Ganter, “Attribute Exploration with Background Knowledge”, Theoretical Computer Science, vol. 217, pp. 215—233.

[5] T. Hidayat, Implicational Bases of Many-Valued Context, Master Thesis, TU-Dresden, Dresden, 2005. unpublished.

[6] G. Stumme, “Attribute Exploration with Background Implications and Exceptions”, Data Analysis and Information Systems. Statistical and Conceptual approaches. Proc. GfKl'95. Studies in Classification, Data Analysis, and Knowledge Organization 7. Springer, Berlin, 1995, pp. 457—469.

[7] R. Belohlavek and V. Vychodil, “Adding Background Knowledge to Formal Concept Analysis via Attribute Dependency Formulas”, SAC. ACM, 2008, pp. 938—943.

[8] B. Armin, etc, Handbook of Satisfiability, IOS Press, Amsterdam. 2009.

[9] J. Gu, P.W. Purdom, J. Franco, and B.W. Wah, “Algorithms for the Satisability(SAT) Problem:ASurvey”, Discrete Mathematics and Theoretical Computer Science, 1996, pp. 19—152.

[10] N. Manthey, Coprocessor - a Standalone SAT Preprocessor, CoRR, vol. abs/1108.6208, 2011.

[11] F. Martinez-Rios and J. Frausto-Solis, “A Hybrid Simulated Annealing and Threshold Accepting for Satisfiability Problems using Dynamically Cooling Schemes”, 6th WSEAS Int. Conference on Computational Intelligence, Man-Machine Systems and Cybernetics, WSEAS Press, 2007, pp. 281—285.

[12] C. Posthoff and B. Steinbach, “SAT-Problems - New Findings”, 9th WSEAS Int. Conference on Data Networks, Communications, Computers, WSEAS Press, 2007, pp. 339—344.

[13] I. Vazquez-Moran and J. Torres-Jimenez, “SAT Instances Construction Based on Hypergraphs”, 3rd WSEAS on NNA, FSFS, EC, WSEAS Press, 2002, pp. 2171—2174.

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

[15] T. Hidayat, “Attribute exploration of many-valued context with SAT”, 10th WSEAS international conference on Computational Intelligence, Man-Machine Systems and Cybernetics, WSEAS Press, 2011, pp. 33—37.

[16] E. Tsang, Foundation of Constraint Satisfaction, Academic Press Limited, London, 1993.