APPLICATION FOR LOGICAL EXPRESSION PROCESSING

Marcin Michalak, Michał Dubiel, Jolanta Urbanek

Institute of Informatics, Silesian University of Technology, Gliwice, Poland
Marcin.Michalak@polsl.pl

ABSTRACT

Processing of logical expressions – especially a conversion from conjunctive normal form (CNF) to disjunctive normal form (DNF) – is very common problem in many aspects of information retrieval and processing. There are some existing solutions for the logical symbolic calculations, but none of them offers a functionality of CNF to DNF conversion. A new application for this purpose is presented in this paper.

KEYWORDS

Boolean Formulas, CNF, DNF, Symbolic Expressions

1. INTRODUCTION

In many aspects of machine learning methods it becomes necessary to represent the knowledge extracted from the data in the terms of a logical formulas. Then the proper tool for Boolean expression processing should be used.

This paper is organized as follows: it starts from a brief overview of problems which solutions are found on the basis of some logical expression analysis; then a short discussion about existing platforms of symbolic formulas processing and their limitation is shown. Afterwards, the new application for CNF to DNF conversion is presented and it's functionality is shown on some short example. The present limitations and the computational complexity of the application are also mentioned. The paper ends with some conclusions and perspectives of the further works.

2. FIELDS OF APPLICATIONS

Many algorithms of machine learning are based on processing information in a logical way. Here two of them are presented: induction of reducts and induction of biclusters.

2.1. Induction of Reducts

A notion of reduct [2] is strongly connected with the theory of data representation – rough sets [1] – defined in the 80's of last century. In this theory data is presented in a table called an information system. Information system is defined as an ordered pair of two non-empty and finite sets: set of objects U and set of attributes A. It is generally assumed that all attributes take only discrete values but there are know some approaches that allows for the continuous attributes in the data.

Natarajan Meghanathan et al. (Eds) : ACSIT, SIPM, CMIT, CoNeCo - 2016
pp. 01–09, 2016. © CS & IT-CSCP 2016 DOI : 10.5121/csit.2016.60801

A sample information system is presented in the Table 1. This information system provides information about four objects, for which the following attributes were measured: temperature, weight, height, color of eyes.

Table 1. A sample information system.

	temperature	**weight**	**height**	**eyes**
U1	high	normal	small	blue
U2	high	low	normal	green
U3	high	low	high	brown
U4	high	high	high	blue

One can notice that the knowledge about values of all attributes for all objects introduced a partition of a set of objects – two objects belong to the same element of partition if they have the same value of the same attributes. With the fundamental theorem on equivalence relation we know that every partition of a set forms an equivalence relation. This relation is called an indiscernibility relation as two object are in a relation when there is no ability to discern each other due to the knowledge about their attributes values.

For a presented information system the indiscernibility relation forms a following partition.

$$U \setminus IND_A = \{\{U_1\}, \{U_2\}, \{U_3\}, \{U_4\}\}$$

The partition formed by this relation is being considered as a knowledge in an information system. It means that any join of elements of partition removes some amount of knowledge from our system. For the information system presented in the Table 1 it is obvious that temperature does not provide any new knowledge as all objects have the same value of this attribute. Excluding the temperature from the set of attributes we obtain the same partition of the set of objects:

$$U \setminus IND_{A\setminus\{temperature\}} = \{\{U_1\}, \{U_2\}, \{U_3\}, \{U_4\}\}$$

This leads us to the question, whether all attributes are necessary to deliver the same knowledge? From this point of view a reduct in an information system is a minimal in the sense of inclusion subset of attributes that introduces the same indiscernibility relation as the whole set A. More formal definition states:

A set $B \subseteq A$ is a reduct for the information system if and only if:

i) $U \setminus IND_A = U \setminus IND_B$

ii) There is no $C \subset B$ that satisfies $U \setminus IND_C = U \setminus IND_B$

Finding reducts is performed with the analysis of the Boolean formula build as follows: for every attribute a Boolean variable is assigned; for every two objects a logical sum of variables corresponding to attribute that differs these two objects is constructed (when two objects are identical no logical sum is built); final formula is an conjunction of alternatives. This formula is called the discernibility function. Finding the discernibility function becomes easier when all alternatives are presented in the discernibility matrix. This matrix has the same number of rows and columns as the number of objects in the information system. Each cell of this matrix contains attributes that differs objects corresponding to the row and the column of the matrix. It becomes from the definition that this matrix is symmetric. A discernibility matrix for the system presented

in the Table 1 has the following form as presented in the Table 2. Logical variables t, w, h, e, correspond to the attributes temperature, weight, height and eyes.

Table 2. Discernibility matrix for the information system.

	U1	**U2**	**U3**	**U4**
U1	$\emptyset$	$\{w, h, e\}$	$\{w, h, e\}$	$\{w, h\}$
U2	$\{w, h, e\}$	$\emptyset$	$\{h, e\}$	$\{w, h, e\}$
U3	$\{w, h, e\}$	$\{h, e\}$	$\emptyset$	$\{w, e\}$
U4	$\{w, h\}$	$\{w, h, e\}$	$\{w, e\}$	$\emptyset$

The discernibility function will take the following initial form (repeating alternatives were omitted):

$$f(t, w, h, e) = (w \vee h \vee e) \wedge (w \vee h) \wedge (h \vee e) \wedge (w \vee e)$$

As we can see the value of the function does not depend on the value of the variable t, corresponding to the temperature attribute. When simplified, according to the absorption rules, and transformed to the disjunctive normal form, the function takes a form:

$$f(t, w, h, e) = (w \wedge e) \vee (w \wedge h) \vee (h \wedge e)$$

which is built from three prime implicants. This means that there are three two-element reducts for the information system. As it was expected temperature is not an element of any of them as it does not provide any new knowledge.

There is a theorem that says that for every reduct in the information system there is a prime implicant of the discernibility function and vice versa. Attributes in a reduct are the attributes corresponding to the variables in the implicant.

From a formal point of view the problem of finding reducts in the information system is a problem of finding prime implicants of a Boolean formula presented as the logical conjunction of alternatives of non-negated variables.

2.2. Induction of Relative Reducts

Let us consider a specific type of an information system with an emphasized attribute called a decision attribute (or just a decision). This kind of information system is called a decision table and is defined as an ordered pair of finite sets of objects and attributes, where one attribute d is denoted as the decision $(U, A \cup \{d\})$. All non-decision attributes are called conditional attributes. Generally, decision tables contain an information about how the value of the decision depends on values of conditional attributes. It is also worth to state a question whether all conditional attributes are necessary to provide the same amount of knowledge? This leads us to the notion of a relative reducts for the decision table.

Let us define the indiscernibility relation build on the set of conditional attributes A. Let us also define a decision class as the subset of objects with the same value of a decision. The positive region of a decision table ($POS_{DT}(A)$) is a union of equivalence classes that are subsets of decision classes. The higher proportion of objects from positive region to all of objects the more knowledge in the decision table (the more objects can be correctly assign to proper classes). Than as a relative reduct a subset of conditional attributes that give the same amount of knowledge as the whole set of conditional attributes can be defined. The formal definition of a relative reduct is as follows:

A set $B \subseteq A$ is a relative reduct for the information system if and only if:

i) $POS_{DT}(B) = POS_{DT}(A)$

ii) There is no $C \subset B$ that satisfies $POS_{DT}(C) = POS_{DT}(B)$

Finding relative reducts can be also done with the analysis of some Boolean formula.This formula is based on an information stored in a matrix called modulo d discernibility matrix. The phrase "modulo d" is due to the fact that we do not want to discern objects from the same decision class. So only information about difference between objects from different decision classes is presented in this matrix. Let there be a following decision table (Table 3).

Table 3 A sample decision table.

	A	**B**	**C**	**d**
1	a	a	a	yes
2	b	a	b	yes
3	b	c	d	no
4	c	c	d	no

A modulo d discernibility matrix for this decision table will take a following form:

Table 4 A modulo d discernibility matrix for a decision table.

	1	**2**	**3**	**4**
1	$\emptyset$	$\emptyset$	$\{a, b, c\}$	$\{a, b, c\}$
2	$\emptyset$	$\emptyset$	$\{b, c\}$	$\{a, b, c\}$
3	$\{a, b, c\}$	$\{b, c\}$	$\emptyset$	$\emptyset$
4	$\{a, b, c\}$	$\{a, b, c\}$	$\emptyset$	$\emptyset$

Due to the absorption rules we obtain a following disjunction form of the formula:

$$f = b \vee c$$

This means that there are two one-attribute relative reducts for this decision table. This also means that the A attribute is completely redundant.

2.3. Induction of Decision Bireducts

A notion of a decision bireduct [3] also refers to the decision table. Let us consider the same decision table as presented in Table 3. It contains the same kind of information as the previous one but derived from a different objects

As it can be easily observed two last objects are identic so they cannot be discerned due to any possible set of attributes. It is worth to be considered whether do there exist some subsets of objects which are completely discernible due to the subset of attribute. A presented information is delivered by a notion of a bireduct, which definition is as follows:

For a decision system represented by an ordered pair $(U, A \cup \{d\})$ of a set of objects U and a set of attributes A a following ordered pair (B, X) of $B \subseteq A$ and $X \subseteq U$ is called an information bireduct if and only if B discerns all pairs of objects in X (i.e. for each $i, j \in X$ there exists at least one $b \in B$ such that objects i and j differs at least on an attribute b and the following properties hold:

1. There is no proper subset $C \subset B$ such that C discerns all pairs in X, where $d(i) \neq d(j)$.

2. There is no proper superset $Y \supset X$ such that B discerns all pairs in Y, where $d(i) \neq d(j)$.

There also exists a construction of a Boolean formula of logical variables corresponding to attributes and objects, which prime implicants correspond to decision bireducts one to each other.

Let $(U, A \cup \{d\})$ be a decision system. Consider the following Boolean formula with variables $\bar{\imath}, i = 1, \ldots, |U|$ and $\bar{u}, u \in A$:

$$f = \bigwedge_{i,j:d(i)\neq d(j)} \left(\bar{\imath} \vee \bar{\jmath} \vee \bigvee_{a:a(i)\neq a(j)} \bar{a} \right)$$

An arbitrary pair $(B, X), B \subseteq A, X \subseteq U$, is a decision bireduct, if and only if the Boolean formula $\bigwedge_{a\in B} \bar{a} \bigwedge_{i\notin X} \bar{\imath}$ is the prime implicants for f. An initial form of f for the decision table that is considered is as follows:

$$f = (1 \vee 3 \vee \mathrm{A} \vee \mathrm{B} \vee \mathrm{C}) \wedge (1 \vee 4 \vee \mathrm{A} \vee \mathrm{B} \vee \mathrm{C}) \wedge (2 \vee 3 \vee \mathrm{B} \vee \mathrm{C}) \wedge (2 \vee 4 \vee \mathrm{A} \vee \mathrm{B} \vee \mathrm{C})$$

and its DNF form:

$$f = (1 \wedge 2) \vee (3 \wedge 4) \vee (2 \wedge \mathrm{A}) \vee (3 \wedge \mathrm{A}) \vee \mathrm{B} \vee \mathrm{C}$$

Two last bireducts are a typical reducts. Two first describes decision classes. A remaining two are decision bireducts.

3. EXISTING SOLUTIONS

3.1 Matlab

Mathworks provides a toolbox for a symbolic operations. It also allows to define a formulas of Boolean variables, to perform some simplifications but it is not possible to process this formulas exactly to the demanded DNF form.

3.2. R Packages

There exist a package in R which make it possible to find prime implicants for a logical statement but it requires a truth table as an input. This truth table defines alternatives of which variables (negated or not) returns true. This cause that the final prime implicants contain also negated variables.

4. APPLICATION DESCRIPTION

Application is developed in Microsoft Visual Studio 2015 and written in C#. It requires a text input, where labels are Boolean variables and logical operators are represented with '*' and '+'. Alternatives are separated with brackets. As the file with a correct content is loaded a text operators '*' and '+' are displayed as a correct logical operators. The application window with a loaded logical expression is presented in the Fig. 1.

Figure 1. The application with a loaded formula.

The result of calculation is presented in the Fig. 2.

Figure 2. The founded solution: a disjunctive normal form of the expression in a conjunctive normal form.

Result formula can be presented in several ways, including the LaTeX mathematical mode. Let us consider a logical formula as follows:

$$f = (\boldsymbol{u1} \vee \boldsymbol{a1} \vee \boldsymbol{a2}) \wedge (\boldsymbol{u1} \vee \boldsymbol{a1} \vee \boldsymbol{a3}) \wedge (u1 \vee a2 \vee a3) \wedge (u2 \vee a1 \vee a3) \wedge \\ \wedge (u2 \vee a2 \vee a3) \wedge (u1 \vee u2 \vee a2) \wedge (u1 \vee u3 \vee a2) \wedge (u1 \vee u2 \vee a3) \wedge \\ \wedge (u1 \vee u3 \vee a3) \wedge (u2 \vee u3 \vee a3)$$

Below the following steps of transformation from CNF to DNF are presented. In the first step each two consecutive alternatives are logically multiplied. Expressions being processed are typed with the bold font and the result is underlined. For a better clarity of presentation only a logical sum is presented with a binary operator:

$$f = \underline{(u1 \vee a1 \vee a2a3)}(u1 \vee a2 \vee a3)(u2 \vee a1 \vee a3)(\boldsymbol{u2} \vee \boldsymbol{a2} \vee \boldsymbol{a3})(\boldsymbol{u1} \vee \boldsymbol{u2} \vee \boldsymbol{a2}) \\ (u1 \vee u3 \vee a2)(u1 \vee u2 \vee a3)(u1 \vee u3 \vee a3)(u2 \vee u3 \vee a3)$$

$$f = (u1 \vee a1 \vee a2a3)(u1 \vee a2 \vee a3)(u2 \vee a1 \vee a3)\underline{(u1a3 \vee u2 \vee a2)}(u1 \vee u3 \vee a2) \\ (u1 \vee u2 \vee a3)(\boldsymbol{u1} \vee \boldsymbol{u3} \vee \boldsymbol{a3})(\boldsymbol{u2} \vee \boldsymbol{u3} \vee \boldsymbol{a3})$$

$$f = (u1 \vee a1 \vee a2a3)(\boldsymbol{u1} \vee \boldsymbol{a2} \vee \boldsymbol{a3})(u2 \vee a1 \vee a3)(u1a3 \vee u2 \vee a2)(\boldsymbol{u1} \vee \boldsymbol{u3} \vee \boldsymbol{a2})$$
$$(u1 \vee u2 \vee a3)(\underline{u1u2 \vee u3 \vee a3})$$

$$f = (u1 \vee a1 \vee a2a3)\big(\underline{u1 \vee u3a3 \vee a2}\big)(\boldsymbol{u2} \vee \boldsymbol{a1} \vee \boldsymbol{a3})(u1a3 \vee u2 \vee a2)$$
$$(\boldsymbol{u1} \vee \boldsymbol{u2} \vee \boldsymbol{a3})(u1u2 \vee u3 \vee a3)$$

$$f = (\boldsymbol{u1} \vee \boldsymbol{a1} \vee \boldsymbol{a2a3})(\boldsymbol{u1} \vee \boldsymbol{u3a3} \vee \boldsymbol{a2})\big(\underline{u2 \vee u1a1 \vee a3}\big)(u1a3 \vee u2 \vee a2)$$
$$(u1u2 \vee u3 \vee a3)$$

$$f = \big(\underline{u1 \vee u3a1a3 \vee a1a2 \vee \sout{u3a2a3} \vee a2a3}\big)(\boldsymbol{u2} \vee \boldsymbol{u1a1} \vee \boldsymbol{a3})(\boldsymbol{u1a3} \vee \boldsymbol{u2} \vee \boldsymbol{a2})$$
$$(u1u2 \vee u3 \vee a3)$$

$$f = (u1 \vee u3a1a3 \vee a1a2 \vee a2a3)\big(\underline{u2 \vee \sout{u1a1a3} \vee u1a1a2 \vee u1a3 \vee a2a3}\big)$$
$$(u1u2 \vee u3 \vee a3)$$

$$f = (\boldsymbol{u1} \vee \boldsymbol{u3a1a3} \vee \boldsymbol{a1a2} \vee \boldsymbol{a2a3})(u2 \vee u1a1a2 \vee u1a3 \vee a2a3)(\boldsymbol{u1u2} \vee \boldsymbol{u3} \vee \boldsymbol{a3})$$

$$f = (u1u2 \vee u1u3 \vee u1a3 \vee \sout{u1u2u3a1a3} \vee u3a1a3 \vee \sout{u3a1a3} \vee \sout{u1u2a1a2} \vee$$
$$\vee u3a1a2 \vee \sout{a1a2a3} \vee \sout{u1u2a2a3} \vee \sout{u3a2a3} \vee a2a3)(u2 \vee u1a1a2 \vee u1a3 \vee a2a3)$$

$$f = (u1u2 \vee u1u3 \vee u1a3 \vee u3a1a3 \vee u3a1a2 \vee a2a3)(u2 \vee u1a1a2 \vee u1a3 \vee a2a3)$$

For the simplification of calculation let the repeating conjunction be bolded:

$$f = (u1u2 \vee u1u3 \vee \boldsymbol{u1a3} \vee u3a1a3 \vee u3a1a2 \vee \boldsymbol{a2a3})(u2 \vee u1a1a2 \vee \boldsymbol{u1a3} \vee \boldsymbol{a2a3})$$

Now the multiplication becomes more easy:

$$f = u1a3 \vee a2a3 \vee (u1u2 \vee u1u3 \vee u3a1a3 \vee u3a1a2)(u2 \vee u1a1a2)$$

$$f = u1a3 \vee a2a3 \vee u1u2 \vee \sout{u1u2a1a2} \vee \sout{u1u2u3} \vee u1u3a1a2 \vee u2u3a1a3 \vee\vee \sout{u1u3a1a2a3}$$
$$\vee u2u3a1a2 \vee \sout{u1u3a1a2}$$

$$f = u1a3 \vee a2a3 \vee u1u2 \vee u1u3a1a2 \vee u2u3a1a3 \vee u2u3a1a2$$

In the Fig. 3. the application window with the loaded formula and result of it conversion to DNF are presented. The obtain solution is equivalent to the solution obtained as the result of step-by-step calculations.

Figure 3. A considered formula in CNF and DNF.

A weak point of the application is the high computational complexity. On the Fig. 4 an average time of ten experiments as the function of a input formula length is presented.

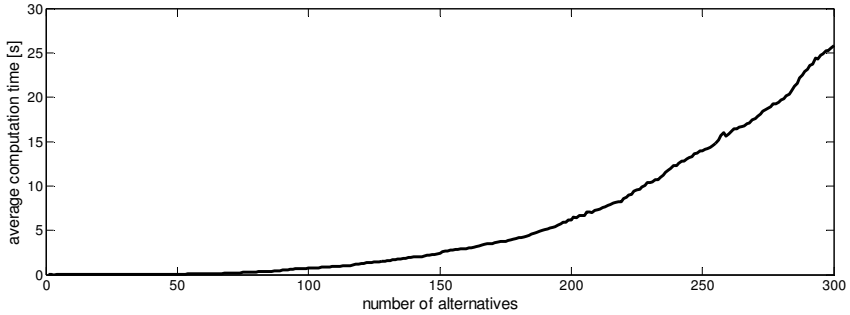

Figure 4. An average time of computation with the increase of the input data length.

As one may observe the computational complexity is polynomial. The degree of the complexity polynomial can be estimated with the Pearson's correlation coefficient between the number of alternatives in the input formula and the appropriate degree of the average computation time root. The results are presented in the Table 1.

Table 1. Correlation between the task complexity and the roots of average time of computation.

Root degree	r
1	0.892489676
2	0.981016972
3	0.998530988
4	0.998626003
5	0.993885497
6	0.987855142
7	0.981611630
8	0.975466952

It can be experimentally estimated that the computational complexity of the algorithm is polynomial with the degree of four.

5. CONCLUSIONS AND FURTHER WORKS

In the paper the new application for logical expression processing is presented. It performs a symbolic Boolean calculations, converting the CNF of the formula to DNF. As results of these computation may be useful in scientific research it provides also a LaTeX format of results. Due to the high computational complexity our further works will focus on reduction of the time of computation.

REFERENCES

[1] Pawlak Z.: Rough Sets. International Journal of Computer and Information Sciences, 11(5):341-356, Warsaw 1982.

[2] Pawlak Z.: Rough Sets. Theoretical Aspects of Reasoning about Data, Springer, 1991

[3] Ślęzak D., Janusz A.: Ensambles of Bireducts: Towards Robust Classification and Simple Representation, Lecture Notes in Computer Science 7105:64-77, 2011

AUTHORS

Marcin Michalak was born in Poland in 1981. He received his M.Sc. Eng. in computer science from the Silesian University of Technology in 2005 and Ph.D. degree in 2009 from the same university. His scientific interests are in machine learning, data mining, rough sets and biclustering. He is an author and coauthor of over 60 scientific papers.

Michał Dubiel is a student at Silesian university of Technology

Jolanta Urbanek received her M. Sc. Eng. in computer science from the Silesian University of Technology. Her scientific interests are in biclustering.