On Application of the Ternary Matrix Cover Technique for Minimization of Boolean Functions

Yu.V. Pottosin, E.A. Shestakov
United Institute of Problems of Informatics, NAS of Belarus, Minsk,
(pott,she)@newman.bas-net.by, tel.:(+375) 17 284 21 64

Abstract: *To solve the task of recognizing features of objects used in expert systems the Boolean function approach can be attracted. In particular, the relation between the features can be given as a disjunctive normal form (DNF) of the Boolean function whose arguments correspond to the features, and the rise of compactness of the information of that relation reduces to minimization of a Boolean function in DNF class. The ternary matrix cover technique is suggested to apply for solving this problem. It is shown how to obtain a minimal set of prime implicants using this technique.*

Keywords: *expert system, prohibition domain, ternary matrix, minimization of Boolean functions.*

I INTRODUCTION

One of the functions of an expert system is recognizing the object by its known features. Any feature is assigned with a Boolean variable that takes value 1 if the object has this feature, and value 0 if it has not [1]. Usually, there is a relation between features that is given by the prohibition domain. This domain is convenient to be given as a disjunctive normal form (DNF) of a Boolean function [1]. The problem of rising compactness of the prohibition domain is reduced to the minimization of a Boolean function in DNF class.

The concept of ternary matrix cover was introduced first while the problem of decomposition of Boolean functions was being studied [2]. It is similar to the concept "blanket" that was used in [3]. The ternary matrix is a form of representation of a completely specified Boolean function. Any ternary matrix can represent some arbitrary DNF of a Boolean function [4]. The rows of ternary matrix U represent the elementary conjunctions that are the terms of DNFs of given functions.

The approach connected with the concept of ternary matrix cover was successfully applied in solving such tasks as decomposition of Boolean functions, revelation of essential arguments and orthogonalization of Boolean functions. We show how one can apply this approach to obtain a minimal family of prime implicants of a Boolean function in solving the problem of its minimization. The problem of selecting the prime implicants that constitute a minimum DNF of the given Boolean function is reduced to the covering problem [4]. We suggest a new way of this reducing based on the operation of product ternary matrix covers. When the input Boolean function is given in arbitrary DNF, this way makes easier the procedure of constructing Quine's table [4] necessary for solving the covering problem.

II TERNARY MATRIX COVER

Let U be a ternary $(l \times n)$-matrix (its elements are 0, 1 and "–") [4], that gives DNF of Boolean function $f(\mathbf{x})$. The columns of matrix U correspond to variables $x_1, x_2, \ldots, x_n$ that are arguments of $f(\mathbf{x})$. Let $\mathbf{x}^* = (x_1^*, x_2^*, \ldots, x_n^*)$ be a value of vector variable $\mathbf{x}$.

A ternary vector $\mathbf{a}$ is said to absorb a binary (Boolean) vector $\mathbf{b}$ if $\mathbf{b}$ can be obtained from $\mathbf{a}$ by replacing symbols "–" by 0 or 1.

A family π of subsets (blocks) of the set of row numbers of ternary $(l \times n)$-matrix U is called cover of U if for any Boolean vector $\mathbf{x}^*$ of length n there exists a block in π containing those and only those rows of U that absorb $\mathbf{x}^*$. Other sets of rows of U are not in π. Let us denote by $t(\mathbf{x}^*, U)$ the set of rows of U that absorb $\mathbf{x}^*$. Then the blocks of cover π are all different sets $t(\mathbf{x}^*, U)$ taken for all $\mathbf{x}^* \in \{0, 1\}^n$. If no row of U absorb a Boolean vector $\mathbf{x}^* \in \{0, 1\}^n$, then one of the blocks of cover π is empty set $\varnothing$.

For every block π_i of cover π, the Boolean function $\pi_i(\mathbf{x})$ is defined such that $\pi_i(\mathbf{x}^*) = 1$ for any $\mathbf{x}^* \in \{0, 1\}^n$ if and only if $t(\mathbf{x}^*, U) = \pi_i$.

For any ternary matrix there exists the single cover.

The methods for calculating the cover of a given ternary matrix are described in [5]. The simplest one of them is based on the operation of product of covers that is determined as follows.

Let covers π^1 and π^2 be constructed for ternary matrices U^1 and U^2 relatively whose numbers of rows are the same and sets of columns correspond to nonempty subsets of the set of variables $x_1, x_2, \ldots, x_n$. Let us form the set

$$\lambda = \{\pi^1_i \cap \pi^2_j / \pi^1_i \in \pi^1, \pi^2_j \in \pi^2_i, \pi^1_i(\mathbf{x}) \wedge \pi^2_j(\mathbf{x}) \neq 0\}$$

and define the Boolean function $\lambda_{ij}(\mathbf{x}) = \pi^1_i(\mathbf{x}) \wedge \pi^2_j(\mathbf{x})$ for every element $\lambda_{ij} = \pi^1_i \cap \pi^2_j$ of λ. Let us construct cover π taking all different element of λ as elements of π. For every block π_k of π we define the Boolean function $\pi_k(\mathbf{x})$ as disjunction of all the functions assigned to elements of λ equal to π_k. We call cover π the product of covers π^1 and π^2 ($\pi = \pi^1 \times \pi^2$). The product of covers is shown in [6] to be commutative, associative and idempotent.

Let a given ternary matrix U be divided into two matrices U^1 and U^2 where U^1 consists of some column of U and U^2 of the rest of the columns of U. If π^1 and π^2 are the covers of matrices U^1 and U^2 relatively, then $\pi = \pi^1 \times \pi^2$ is the cover of matrix U. The cover of an one-column matrix is trivial. It has two blocks, one of which consists of rows containing 0 and "–", the other of rows containing "–" and 1. The functions assigned to them are $\bar{x}$ and x relatively where x is the variable corresponding to the considered column. So, cover π of matrix U can be obtained as $\pi = \pi^1 \times \pi^2 \times \ldots \times \pi^n$ where $\pi^1, \pi^2, \ldots, \pi^n$ are the covers of one-column matrices equal to the columns of U connected with variables $x_1, x_2, \ldots, x_n$ relatively.

III OBTAINING A MINIMAL FAMILY OF PRIME IMPLICANTS OF A BOOLEAN FUNCTION

Let a ternary matrix U represent a reduced DNF of a Boolean function $f(\boldsymbol{x})$, i. e. its rows represent the prime implicants of $f(\boldsymbol{x})$. The minimal DNF of function $f(\boldsymbol{x})$ is the disjunction of a minimum number of prime implicants such that for every value $\boldsymbol{x}^*$ of vector variable $\boldsymbol{x}$ at which $f(\boldsymbol{x}) = 1$, there is at least one among them that is equal to 1 at $\boldsymbol{x} = \boldsymbol{x}^*$.

A minimal DNF of Boolean function $f(\boldsymbol{x})$ is represented by a ternary matrix that consists of a minimal family of rows of matrix U covering all values of vector variable $\boldsymbol{x}$ that turn $f(\boldsymbol{x})$ into 1. So, the problem is reduced to the classical problem of covering: to find a minimal subset of the set of matrix U such that for every nonempty block π_i of its cover π there is a row in this subset that belong to block π_i. In the terminology of [7] every block of such a cover is a Quine's set.

Indeed, as it was said, every block π_i of cover π of ternary matrix U is equal to a set $t(\boldsymbol{x}^*, U)$ of rows of matrix U that absorb a certain vector $\boldsymbol{x}^*$, and for every vector $\boldsymbol{x}^*$ there is a block in π containing the row of U that absorb $\boldsymbol{x}^*$. So, every value $\boldsymbol{x}^*$ such that $f(\boldsymbol{x}^*) = 1$ is absorbed at least by one row from the set satisfying the condition above.

The complexity of the problem of covering can be reduced if there are obligatory prime implicants, i. e. those being in any minimal and even irredundant DNF. The set of such implicants is called kernel [4]. The whole kernel is included in the obtained DNF, and all values $\boldsymbol{x}^*$ of vector variable $\boldsymbol{x}$ covered by the kernel are excluded from the consideration. The prime implicants absorbed by the kernel form so called anti-kernel [4]. They must be excluded from the consideration as well.

It is clear that if block π_i of cover π of matrix U consists only of one row, then this row represents a prime implicant belonging to the kernel. It is easy to extract the kernel from matrix U if its cover is obtained.

The prime implicants corresponding to the rows of U each of which is only in the blocks of π where there is at least one element of the kernel form the anti-kernel.

In minimizing a Boolean function, any block of cover π containing another block as a subset may be excluded from the consideration. This and above acts of reducing agree with reduction rules [4] that are used in solving the problem of covering.

Example

Let us consider the matrix of prime implicants taken from [7]:

$$U = \begin{array}{c} \begin{array}{ccccc} x_1 & x_2 & x_3 & x_4 & x_5 \end{array} \\ \left[\begin{array}{ccccc} 1 & 0 & - & 1 & 1 \\ 0 & 0 & 0 & - & 1 \\ 0 & - & 1 & 0 & - \\ - & 0 & 0 & 1 & 1 \\ 0 & 0 & - & 0 & 1 \\ 1 & - & 1 & 1 & 1 \\ - & - & 1 & 0 & 0 \\ 1 & 1 & - & 0 & - \\ - & 1 & 1 & 0 & - \\ - & 1 & 1 & - & 1 \end{array}\right] \end{array} \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{array}$$

For one-column matrices obtained from U we have two-block covers $\pi^1 = \{(1, 4, 6, 7, 8, 9, 10), (2, 3, 4, 5, 7, 9, 10)\}$, $\pi^2 = \{(1, 2, 3, 4, 5, 6, 7), (3, 6, 7, 8, 9, 10)\}$, $\pi^3 = \{(1, 2, 4, 5, 8), (1, 3, 5, 6, 7, 8, 9, 10)\}$, $\pi^4 = \{(1, 2, 4, 6, 10), (2, 3, 5, 7, 8, 9, 10)\}$, $\pi^5 = \{(1, 2, 3, 4, 5, 6, 8, 9, 10), (3, 7, 8, 9)\}$.

The product of them is $\pi = \pi^1 \times \pi^2 \times \pi^3 \times \pi^4 \times \pi^5 = \{(1, 4), (1, 6), (7), (8), (6, 10), (7, 8, 9), (8, 9, 10), (2, 4), (2, 5), (3, 5), (3, 7), (3, 7, 9), (3, 9, 10), (10)\}$ that is the cover of matrix U.

Rows 7, 8 and 10 form one-element blocks that are elements of the kernel. The anti-kernel consists of only one row 9, as it is the only row that is only in those blocks that contain the elements of the kernel. Having kept in the obtained cover π only those blocks that do not contain other blocks as subsets, we obtain $\{(1, 4), (1, 6), (7), (8), (2, 4), (2, 5), (3, 5), (10)\}$. Remove one-element sets from the obtained family, and as a result of it we have $\{(1, 4), (1, 6), (2, 4), (2, 5), (3, 5)\}$. Then, we should choose a minimal set of rows of U such that any of remaining blocks contains at least one row of this set.

One of the solutions of our task is the matrix consisting of rows 1, 2, 5, 7, 8, 10 of matrix U:

$$\begin{array}{c} \begin{array}{ccccc} x_1 & x_2 & x_3 & x_4 & x_5 \end{array} \\ \left[\begin{array}{ccccc} 1 & 0 & - & 1 & 1 \\ 0 & 0 & 0 & - & 1 \\ 0 & 0 & - & 0 & 1 \\ - & - & 1 & 0 & 0 \\ 1 & 1 & - & 0 & - \\ - & 1 & 1 & - & 1 \end{array}\right] \end{array} \begin{array}{c} \\ 1 \\ 2 \\ 5 \\ 7 \\ 8 \\ 10 \end{array}$$

Note, that the field of the search can be considerably reduced, if in successive taking products of covers, the rows forming one-element blocks are removed.

IV CONCLUSION

The described approach is intended to apply it in solving the problems of the Boolean function theory that is connected with the specification of functions in the form of a ternary matrix representing DNF. It should be noted that efficiency of this approach depends very largely on the number of blocks of the cover of the initial ternary matrix. The suggested approach can have the advantage compared with the method of simple sets described in [4, 7] for ternary matrices with the relatively large number of rows and small number of columns.

The work is partly supported by ISTC, Project B-986.

REFERENCES

[1] Zakrevskii A.D. Logic of Recognition. – Moscow: URSS, 2003. – 140 p. (in Russian).

[2] Shestakov E. Decomposition of systems of completely defined Boolean functions by argument covering // Automatic Control and Computer Science. – 1994. – vol. 28. – No. 1. – P. 12-20.

[3] Brzozowski J.A., Luba T. Decomposition of Boolean functions specified by cubes. Research Report CS-97-01, University of Waterloo, Waterloo, Canada, 1997; REVISED October 1998. – 36 p. (ftp://cs-archive.uwaterloo.ca/cs-archive/CS-97-01/CS-97-01.ps.Z).

[4] Zakrevskii A.D. Logical Design of Cascaded Circuits. – M.: Nauka, 1981. – 416 p. (in Russian).

[5] Pottosin Yu.V., Shestakov E.A. Orthogonalization of a system of completely specified Boolean functions // Logical Design. Issue 5. – Minsk: Institute of Engineering Cybernetics, NAS of Belarus, 2000. – P. 107-115 (in Russian).

[6] Pottosin Yu., Shestakov E. Decomposition of systems of completely specified Boolean functions using their compact table representation // Boolean Problems. 4th International Workshop. – Freiberg (Sachen), 2000. – P. 135–142.

[7] Zakrevskii A.D., Pottosin Yu.V., Cheremisinova L.D. Fundamentals of Logical Design. Book 2. Optimization in Boolean Space. – Minsk: United Institute of Informatics Problems, NAS of Belarus, 2004. – 240 p. (in Russian).