Logic synthesis of n-ary quantitative relations

Grzegorz Łabiak

***Abstract*—This contribution shows how to by means of Boolean equation express mathematical abstract notion of n-ary relation. Main criterion of the relation is quantitative criterion, where domain sets of the relation can be compared just like ordinal numbers.**

***Keywords*—logic synthesis, binary relation, n-ary relation, quantitative relations**

I. Introduction

The notion of relation functions in everyday life, computer science, logic and philosophy and in mathematics. Especially in mathematics, relations are one of many fundamental objects which as a part of mathematic metalanguage serve as a means to describe other mathematical notions [4]. The most typical case of relation is a binary relation, where relation is defined on two sets. An n-ary relation is a generalized binary relation defined on n sets. The quantitative relation is a relation where criterion of belongings to the relation is a number of ones in an element of the set. For example, if the set is a set of minterms, then quantity criterion is a number of logic ones in a minterm.

Logic synthesis is a process by which an abstract form of a system (eg. digital circuit behavior) is turned into Boolean expressions suitable for physical implementation (eg. Register Transfer Level) [1]. Main goal of the paper is to present an algorithm how to transform abstract notion of n-ary relation into Boolean expressions with the view of implementation it by means of Binary Decision Diagrams [3][5] in computer memory. Then, physically implemented relations can be part of a bigger and more complex system which performs sophisticated formal analyses [2].

II. The sets and its characteristic functions

Boolean function is a function of the form $f : \mathrm{B}^k \rightarrow \mathrm{B}$, where $\mathrm{B} = \{0, 1\}$ and k is a non-negative integer value called arity of the function. Boolean function can be expressed as propositional formula in k variables. One of most popular canonical form of propositional formula is a form of *Sum of products* called minterms, where formula sums only these minterms which give logic one. Minterm is a product term of k variables $x_1, x_2, \ldots, x_k$ in which each of k variables appears once in either its complemented or uncomplemented form.

Definition 1 (Characteristic function): The *characteristic Boolean function* χ_A of a set of elements $A \subseteq U$ is a Boolean function: $\chi_A : U \rightarrow \{0, 1\}$ defined as follows:

$$\chi_A(x) = \begin{cases} 1 & \iff x \in A \ , \\ 0 & \text{otherwise.} \end{cases}$$

G. Łabiak is with Computer Engineering & Electronics Department, University of Zielona Góra, Licealna 9, 65-417 Zielona Góra, Poland, e-mail: G.Labiak@iie.uz.zgora.pl.

In other words characteristic Boolean function defines a set, where elements of the set are minterms which evaluate to logic one.

III. Binary relation

Definition 2 (Binary relation): The binary relation R between two sets A and B is an ordered triple (A, B, f^2), where f^2 is a Boolean function. The set A is called domain, the set B is called codomain of the relation R and f^2 is a function which on Cartesian product $A \times B$ gives Boolean value.

The set A and B can be arbitrary sets and the function $f^2(x, y)$, where $x \in X$ and $y \in Y$, is a characteristic function of the relation R defined on a Cartesian product $A \times B$. The statement $(x, y) \in R$, where $R \subseteq A \times B$, means that x relates to y, and is denoted by xRy or $R(x, y)$ iff the characteristic function $f^2(x, y)$ of the relation R gives *true* value.

The criterion of the relation is a number of ones in an element of the set. For example, if we have vectors `1110` and `0011` the former is greater then the latter, for the vectors `1010` and `0101`, as a elements of the set, the relation between them is equal. As we see the characteristic function of the relation is a simple logic function which can be constructed in traditional way, eg. using Kanaugh maps. Taking account quantitative relations we can consider three fundamental cases: *is greater then*, *is equal to* and *is greater or equal to*.

A. Is greater then

Let us construct a logic function composed of two inputs each of which consist of three bits. The function gives *true* when number of ones in the first input (variables f, e, d) is greater then number of ones in the second input (variables a, b, c). The function in disjunctive normal form is as follows:

$$\begin{aligned} f^2_{fed>cba} &= \Sigma(8, 16, 24, 25, 26, 28, 32, 40, 41, 42, 44, \\ &\quad 48, 49, 50, 52, 56, 57, 58, 59, 60, 61, 62) \end{aligned}$$

After minimization and other transformations we have:

$$\begin{aligned} f^2_{fed>cba} &= \underbrace{def}_{=3}\,\underbrace{\overline{(abc)}}_{<3} + \underbrace{(de + df + ef)}_{\geq 2}\,\underbrace{\overline{(ab + ac + bc)}}_{<2} + \\ &+ \underbrace{(d + e + f)}_{\geq 1}\,\underbrace{\overline{(a + b + c)}}_{<1} \end{aligned}$$

The structure of the expression explain its functional meaning. The expression is a sum of products and each of the three products computes the case of the given number of bits on input. For example, first product $def * \overline{(abc)}$ gives *true* when on first input number of ones equals 3 (def) and number of ones on the second input does not equal 3 ($\overline{abc}$). First factor of the second product $de + df + ef$ gives *true* when number of ones on first input is greater then or equal 2, but second factor

$\overline{ab+ac+bc}$ gives *true* when number of ones on second input is lest than 2 (is not greater then or equal 2). The number of terms in the factor is a number of combinations $\binom{n}{2}$, where n is number of comparable bits.

Generalized form of the function comparing two n-bits minters is as follows:

$$f^2_{>} = \sum_{k=1}^{n-1}\left(\sum_{c\in\binom{\mathrm{supp}A}{k}}\prod_{i=1}^{k}c_i * \overline{\sum_{c\in\binom{\mathrm{supp}B}{k}}\prod_{i=1}^{k}c_i}\right)$$

where $\mathrm{supp}A$ (and respectively $\mathrm{supp}B$) is a support of characteristic function of the set A (B), $\binom{\mathrm{supp}A}{k}$ (and respectively $\binom{\mathrm{supp}B}{k}$)) is the set of all k-elements combinations from the set of support variables of the set A (and the set B), c is a single combination, c_i is i-th variable of the c and n is a cardinality of the sets $\mathrm{supp}A$ and $\mathrm{supp}B$. The cardinality of the sets $\mathrm{supp}A$ and $\mathrm{supp}B$ are equal.

B. Is equal to

Now, like in previous subsection, let us construct a function which gives *true* when number of bits on both inputs equals. The function in disjunctive normal form is as follows:

$$\begin{aligned} f^2_{fed=cba} &= \Sigma(0,9,10,12,17,18,20,27,29,30,\\ &\quad 33,34,36,43,45,46,51,53,54,63) \end{aligned}$$

After minimization and transformation we have:

$$\begin{aligned} f^2_{fed=cba} &= d\odot a*e\odot b*f\odot c+\\ &\quad d\odot a*e\odot c*f\odot b+\\ &\quad d\odot b*e\odot a*f\odot c+\\ &\quad d\odot b*e\odot c*f\odot a+\\ &\quad d\odot c*e\odot a*f\odot b+\\ &\quad d\odot c*e\odot b*f\odot a+ \end{aligned}$$

In case of comparing 3-bits minterms we must check bits pairwise, bits from first input are checked (operator $\odot$ – EX-OR) against bits from second input whether number of ones are equal, hence we have product of three pairs. We must check every possible triple of pairs. The number of triples is a number of permutations of variables which are support of the set B and equals $3! = 6$.

Generalized form of the function is as follows:

$$f^2_{=} = \sum_{\pi_B\in S(\mathrm{supp}B)}\prod_{\substack{i=1\\ \pi_A(\mathrm{supp}A)}}^{n}\pi_A(i)\odot\pi_B(i)$$

where $S(\mathrm{supp}B)$ is the set of all permutations of the set $\mathrm{supp}B$, $\pi_A(\mathrm{supp}A)$ is an any permutation of the support of the characteristic function of the set A, π_B is a single permutation of the set $\mathrm{supp}B$, $\pi_B(i)$ is a i-th variable of the permutation of $\mathrm{supp}B$ ($\pi_A(i)$ is a i-th variable of the permutation of $\mathrm{supp}A$).

C. Is greater or equal to

This function gives *true* when number of bits on first input is greater or equal to number of bits on second input. Disjunctive normal form of the function is as follows:

$$\begin{aligned} f^2_{fed\geq cba} &= \Sigma(0,8,9,10,12,16,17,18,20,24,25,26,27,\\ &= 28,29,30,32,33,34,36,40,41,42,43,44,45,\\ &= 46,48,49,50,51,52,53,54,55,56,57,58,58,\\ &= 60,61,62,63) \end{aligned}$$

After transformation we have:

$$\begin{aligned} f^2_{fed\geq cba} &= \underbrace{def}_{=3}+\underbrace{(de+df+ef)}_{\geq 2}\underbrace{\overline{(abc)}}_{<3}+\\ &+\ \underbrace{(d+e+f)}_{\geq 1}\underbrace{\overline{(ab+ac+bc)}}_{<2}+\underbrace{\overline{(a+b+c)}}_{<1} \end{aligned}$$

Generalized form is as follows:

$$f^2_{\geq} = \sum_{k=0}^{n-1}\left(\sum_{c\in\binom{\mathrm{supp}A}{k+1}}\prod_{i=1}^{k}c_i * \overline{\sum_{c\in\binom{\mathrm{supp}B}{k}}\prod_{i=1}^{k}c_i}\right)$$

IV. n-ARY RELATION

Definition 3 (n-ary relation): The n-ary relation R between the sets $A_1, A_2, \ldots, A_n$ is an ordered n+1-tuple $(A_1, A_2, \ldots, A_n, f^n)$, where f^n is a Boolean function. The f^n is a function defined on Cartesian product $A_1\times A_2\times\ldots\times A_n$ which gives Boolean value.

Similarly to binary relation, the sets $A_1, A_2, \ldots, A_n$ can be arbitrary sets and the function $f^n(a_1, a_2, \ldots, a_n)$, where $a_1 \in A_1$ and $a_2\in A_2$ and so on till $a_n\in A_n$, is a characteristic function of the relation R defined on a Cartesian product $A_1\times A_2\times\ldots\times A_n$. The statement $(a_1, a_2, a_3, \ldots, a_n)\in R$ means that a_1 relates to a_2 and relates to a_3 and so on till a_n, and is denoted by $a_1Ra_2Ra_3R\ldots Ra_n$ or $R(a_1, a_2, , a_3, \ldots, a_n)$ iff the characteristic function $f^n(a_1, a_2, a_3, \ldots, a_n)$ of the relation R gives *true* value.

n-ary relation can be computed as a simple product of binary relations, namely characteristic Boolean function of n-ary relation is a product of characteristic Boolean functions of its component binary relations:

$$\begin{aligned} &f^n(a_1, a_2, a_3, \ldots, a_{n-1}, a_n) =\\ &= f^2(a_1,a_2)*f^2(a_2,a_3)*\ldots*f^2(a_{n-1},a_n) \end{aligned}$$

REFERENCES

[1] G. de Micheli, *Synteza i optymalizacja układów cyfrowych.* Warszawa: Wydawnictwa Naukowo-Techniczne, 1998.

[2] G. Łabiak, "Solving transition conflicts in hierarchical controllers by means of function predicates," in *Boolean problems : 10th international workshop; ISBN: 978-3-86012-438-3.* Freiberg, Niemcy: Freiberg, Technische Universität Bergakademie, 2012, pp. 73–78.

[3] S.-I. Minato, *Binary Decision Diagrams and Applications for VLSI CAD.* Boston: Kluwer Academic Publishers, 1996.

[4] H. Rasiowa, *Wstęp do matematyki współczesnej.* Warszawa: Wydawnictwo Naukowe PWN, 1998.

[5] F. Somenzi, "CUDD: CU Decision Diagram Package Release 2.4.0." WWW, Feb. 2004, department of Electrical and Computer Engineering University of Colorado at Boulder. [Online]. Available: http://vlsi.colorado.edu/ fabio/CUDD/cuddIntro.html