# A First Approach in the Implementation of the Simplification Method of Boolean and Fuzzy Formulas Based on Finite Algebras

José Jairo Soriano Méndez
Omar Salazar Morales
*Engineering Department*
*Universidad Distrital Francisco José de Caldas*
*Bogotá D. C., Colombia*
*Emails: jairosoriano@udistrital.edu.co*
*osalazarm@correo.udistrital.edu.co*

Julián Barreiro Gómez
Miguel Eduardo Hernández Figueredo
*Engineering Department*
*Universidad Santo Tomás*
*Bogotá D. C., Colombia*
*Emails: 2060093@usantotomas.edu.co*
*2060090@usantotomas.edu.co*

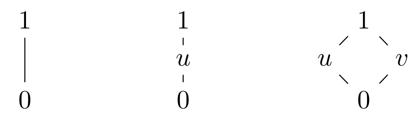
***Abstract*—In several aspects of the fuzzy sets theory and its applications, it is convenient to do some manipulations on the formulas that are proposed in order to obtain some enhancements in methods such as in design and implementation. Currently, the proper fuzzy sets characteristics do not let Boolean simplification methods be applied in an effective way, and consequently, the algorithm solutions proposed so far for these methods cannot be used. This paper presents preliminary ideas in the first algorithmic implementation design of a simplification method of Boolean and fuzzy formulas by using finite algebras.**

***Keywords*-formula manipulation; fuzzy sets; algebra; algorithms;**

## I. Introduction

When Lofti Zadeh [1] introduced the fuzzy set theory also proposed the way to work with sets using their membership functions. The $\max$, $\min$ and $a' = 1 - a$ operations, with $a \in [0,1]$, given by Zadeh have been called *standard operations* [2] because they have been the reference to propose other operations now known as *t-conorms*, *t-norms* and *fuzzy complements*. Some authors [3][4][5] studied the algebraic structures that arose when the standard operations were used over the elements in the unit interval $[0,1]$ and closed intervals $[a,b] \subseteq [0,1]$, and they concluded that they were Kleene and De Morgan algebras. The first case arose when type-1 fuzzy sets were operated, and the second case when interval-valued type-2 fuzzy sets were operated. Now these algebras are known as *standard fuzzy algebras* and they are represented as $\mathcal{K}_s = \langle [0,1]\,; \vee, \wedge, ', 0, 1\rangle$, where $a \vee b = \max\{a,b\}$, $a \wedge b = \min\{a,b\}$ and $a' = 1 - a$ for all $a, b \in [0,1]$; and $\mathcal{M}_s = \langle M; \vee, \wedge, ', 0, 1\rangle$, where $M = \{\langle a,b\rangle \mid a, b \in [0,1]\,, a \leq b\}$, $\langle a,b\rangle \vee \langle c,d\rangle = \langle a \vee c, b \vee d\rangle$, $\langle a,b\rangle \wedge \langle c,d\rangle = \langle a \wedge c, b \wedge d\rangle$ and $\langle a,b\rangle' = \langle 1-b, 1-a\rangle$ for all $\langle a,b\rangle\,, \langle c,d\rangle \in M$. In $\mathcal{M}_s$ an interval $[a,b]$ is represented as a pair $\langle a,b\rangle$ [6][3][4].

Boolean algebra $\mathcal{B}_2 = \langle \{0,1\}\,; \vee, \wedge, ', 0, 1\rangle$, where $\{0,1\}$ is the lattice shown in Figure 1(a) and the operations are shown in table I(a), has several methods for simplification of



(a) (b) (c)

Figure 1. (a) Lattice $\mathcal{B}_2$. (b) Lattice $\mathcal{K}_3$. (c) Lattice $\mathcal{M}_4$.

its formulas. The most known in the literature are Karnaugh maps and Quine-McCluskey method [7]. The problem with these methods is that they are based on boolean properties and they are not valid if they are used in a Kleene or De Morgan algebra. Then, we have a problem if we try to use boolean methods to simplify formulas that involve type-1 fuzzy sets and interval-valued type-2 fuzzy sets (i.e., we are using standard fuzzy algebras). Some authors [8][9][10] have proposed simplification methods in the type-1 fuzzy case, i.e. in the algebra $\mathcal{K}_s$. Their methods are based on the direct manipulation of the formula but not necessarily suitable for computational implementation. In the interval-valued type-2 fuzzy case ($\mathcal{M}_s$ algebra) we have not found proposes.

For the treated subjected in this paper, the mathematical background came in 2003 with the presentation of the algebraic theory by M. Gehrke, C. Walker and E. Walker [4] with some previous works [6][3] where the formulas both in the type-1 and interval-valued type-2 fuzzy cases (i.e. $\mathcal{K}_s$ and $\mathcal{M}_s$ algebras) can be manipulated without losing the identity by using the finite algebras $\mathcal{K}_3 = \langle \{0,u,1\}\,; \vee, \wedge, ', 0, 1\rangle$ and $\mathcal{M}_4 = \langle \{0,u,v,1\}\,; \vee, \wedge, ', 0, 1\rangle$, where $\{0,u,1\}$ and $\{0,u,v,1\}$ are the lattices shown in figures 1(b) and 1(c) and the operations are shown in tables I(b) and I(c). Now is well known that $\mathcal{K}_3$ is a Kleene algebra and $\mathcal{M}_4$ is a De Morgan algebra [4][3][11]. Between 2010 and 2011, Salazar [12] and Soriano, used this algebraic theory to propose a method to simplify the formulas of the algebras $\mathcal{B}_2$, $\mathcal{K}_s$ and $\mathcal{M}_s$ using the finite algebras $\mathcal{B}_2$, $\mathcal{K}_3$ and $\mathcal{M}_4$.

978-0-7695-4500-4/11 $26.00 © 2011 IEEE
DOI 10.1109/ISCID.2011.120

Table I
OPERATIONS IN (a) $\mathcal{B}_2$, (b) $\mathcal{K}_3$ AND (c) $\mathcal{M}_4$.

(a)

| $\vee$ | 0 | 1 |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 1 | 1 |

| $\wedge$ | 0 | 1 |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |

| $'$ | |
|---|---|
| 0 | 1 |
| 1 | 0 |

(b)

| $\vee$ | 0 | $u$ | 1 |
|---|---|---|---|
| 0 | 0 | $u$ | 1 |
| $u$ | $u$ | $u$ | 1 |
| 1 | 1 | 1 | 1 |

| $\wedge$ | 0 | $u$ | 1 |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| $u$ | 0 | $u$ | $u$ |
| 1 | 0 | $u$ | 1 |

| $'$ | |
|---|---|
| 0 | 1 |
| $u$ | $u$ |
| 1 | 0 |

(c)

| $\vee$ | 0 | $u$ | $v$ | 1 |
|---|---|---|---|---|
| 0 | 0 | $u$ | $v$ | 1 |
| $u$ | $u$ | $u$ | 1 | 1 |
| $v$ | $v$ | 1 | $v$ | 1 |
| 1 | 1 | 1 | 1 | 1 |

| $\wedge$ | 0 | $u$ | $v$ | 1 |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| $u$ | 0 | $u$ | 0 | $u$ |
| $v$ | 0 | 0 | $v$ | $v$ |
| 1 | 0 | $u$ | $v$ | 1 |

| $'$ | |
|---|---|
| 0 | 1 |
| $u$ | $u$ |
| $v$ | $v$ |
| 1 | 0 |

Salazar also showed that is possible the simplification when the standard complement operation $a' = 1 - a$ is changed for one complement operation that satisfies two axiomatic requirements: *monotonously decreasing* and *involutive*, i.e., if $a, b \in [0, 1]$ and $a \leq b$ then $a' \geq b'$ and $(a')' = a$. These algebras were called *quasi-standard fuzzy algebras*, represented as $\mathcal{K}_{qs}$ and $\mathcal{M}_{qs}$, and they were presented in [13].

The exact method proposed by Salazar is not shown in this paper, because its description is too long and we have a restriction in pages. We omit several details. All the mathematical theory can be found in [12]. The aim of this paper is to present some preliminaries ideas behind a first computational implementation made by Barreiro and Hernández. This work is now in progress.

This paper is organized as follows: In Section II we give a summary of the method proposed in [12]. In Section III we discuss preliminaries ideas behind the computational implementation of the method. In Section IV we present the conclusion.

## II. SUMMARY OF THE METHOD

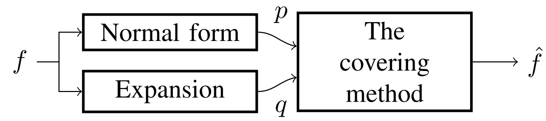
In this summary we omit several details. Given a formula $f$ with $n$ variables in the set $V = \{x_1, \ldots, x_n\}$ we must do the following (see Figure 2):

1) Find the disjunctive $\mathrm{D}(f)$ or conjunctive $\mathrm{C}(f)$ normal form according to the algebra ($\mathcal{B}_2$, $\mathcal{K}_3$ or $\mathcal{M}_4$). The exact rules to find $\mathrm{D}(f)$ and $\mathrm{C}(f)$ using tables of values in $\{0, 1\}$, $\{0, u, 1\}$ or $\{0, u, v, 1\}$ are given in [4]. The disjunctive normal form $\mathrm{D}(f)$ is a representation

$$\mathrm{D}(f) = g_0 \vee g_1 \vee \cdots \vee g_{k-2} \vee g_{k-1}$$

where $g_i$ $(0 \leq i \leq k-1)$ are conjunctions, and the conjunctive normal form $\mathrm{C}(f)$ is a representation

$$\mathrm{C}(f) = h_0 \wedge h_1 \wedge \cdots \wedge h_{l-2} \wedge h_{l-1}$$




$p = \mathrm{D}(f)$ and $q = \mathrm{ED}^V(f)$; or
$p = \mathrm{C}(f)$ and $q = \mathrm{EC}^V(f)$

Figure 2. Block diagram of the simplification method

where $h_i$ $(0 \leq i \leq l-1)$ are disjunctions.

2) Find the disjunctive $\mathrm{ED}^V(f)$ or $\mathrm{EC}^V(f)$ conjunctive expansion according to the algebra ($\mathcal{B}_2$, $\mathcal{K}_3$ or $\mathcal{M}_4$), depending on previously we find $\mathrm{D}(f)$ o $\mathrm{C}(f)$. The exact rules to find $\mathrm{ED}^V(f)$ and $\mathrm{ED}^V(f)$ using tables of values in $\{0, 1\}$, $\{0, u, 1\}$ or $\{0, u, v, 1\}$ are given in [12]. The disjunctive expansion $\mathrm{ED}^V(f)$ is a representation

$$\mathrm{ED}^V(f) = p_0 \vee p_1 \vee \cdots \vee p_{m-2} \vee p_{m-1}$$

where $p_i$ $(0 \leq i \leq m-1)$ are conjunctions, and the conjunctive expansion $\mathrm{EC}^V(f)$ is a representation

$$\mathrm{EC}^V(f) = q_0 \wedge q_1 \wedge \cdots \wedge q_{r-2} \wedge q_{r-1}$$

where $q_i$ $(0 \leq i \leq r-1)$ are disjunctions.

3) We must compare the normal form with the expansion using the covering method. The covering method is a tabular array as shown in tables II(a) and II(b), depending on whether we are using $\langle \mathrm{D}(f), \mathrm{ED}^V(f) \rangle$ or $\langle \mathrm{C}(f), \mathrm{EC}^V(f) \rangle$ respectively.
In table II(a) we put the symbol "✓" if the literals (a variable or its complement) of the conjunction $p_j$ also are inside the conjunction $g_i$. Otherwise we omit the symbol "✓".
In table II(b) we put the symbol "✓" if the literals of the disjunction $q_j$ also are inside the conjunction $h_i$. Otherwise we omit the symbol "✓".
In this point we find at least one formula $\hat{f}$ that satisfies a simplification criterion (this formula is not necessarily unique). The possible formula is given by a subset of $p_j$'s or $q_j$'s such that the symbol "✓" *covers* all the $g_i$'s or $h_i$'s.

## III. ALGORITHM OF SIMPLIFICATION

Once the theory has been proposed, the necessity to implement the mathematical procedure in an algorithm (see Figure 3) came up since it turns cumbersome when increasing the number of variables.

The first step is to get the formula that is required to simplify into the algorithm. This procedure could be given in a table in which the user assigns the values for each one of the combinations according to the algebra and number of variables, or by typing the formula using the possible

Table II
TABULAR ARRAY OF THE COVERING METHOD. (a) ARRAY WITH $\mathrm{D}(f)$ AND $\mathrm{ED}^V(f)$. (b) ARRAY WITH $\mathrm{C}(f)$ AND $\mathrm{EC}^V(f)$.

(a)

| | $\mathrm{ED}^V(f)$ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | $f$ | $p_0$ | $p_1$ | $\cdots$ | $p_j$ | $\cdots$ | $p_{m-2}$ | $p_{m-1}$ |
| $\mathrm{D}(f)$ | $g_0$ | | | | | | | |
| | $g_1$ | | | | | | | |
| | $\vdots$ | | | | | | | |
| | $g_i$ | | | | ✓ | | | |
| | $\vdots$ | | | | | | | |
| | $g_{k-2}$ | | | | | | | |
| | $g_{k-1}$ | | | | | | | |

(b)

| | $\mathrm{EC}^V(f)$ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | $f$ | $q_0$ | $q_1$ | $\cdots$ | $q_j$ | $\cdots$ | $q_{r-2}$ | $q_{r-1}$ |
| $\mathrm{C}(f)$ | $h_0$ | | | | | | | |
| | $h_1$ | | | | | | | |
| | $\vdots$ | | | | | | | |
| | $h_i$ | | | | ✓ | | | |
| | $\vdots$ | | | | | | | |
| | $h_{l-2}$ | | | | | | | |
| | $h_{l-1}$ | | | | | | | |

operations in these algebras ($\wedge$, $\vee$, $'$) as shown above. Either from the table or the formula previously mentioned, this information is saved into an array in order to make all of the operations; mainly the normal form and the expansion after having selected the disjunctive or conjunctive way. The former is found from the array that describes the initial formula, the possible combinations for a certain number of variables and based on the rules of the normal form chosen; whereas the disjunctive/conjunctive expansion is found from the possible combinations including the omission of the variables. Afterwards, the results are manipulated to express the normal form and the expansion as symbolic formulas such as the disjunction of conjunctions or the conjunction of disjunctions depending on the selected way for the procedure.

The covering process, which is the penultimate one, can now be done, since the normal form and the expansion have already been found. However, prior to doing this step, the forms can be simplified by using the absorption property. The absorption property is applied by locating the expressions which have fewer literals than the others, and then looking for those expressions which contain them and are composed by more literals at the same time; so they are despised to rewrite the normal form and the disjunctive/conjunctive expansion.

Finally, it is possible to create a matrix whose columns are the literals obtained in the absorption of the disjunctive/conjunctive expansion and whose rows are the literals obtained in the absorption of the normal form; then, the

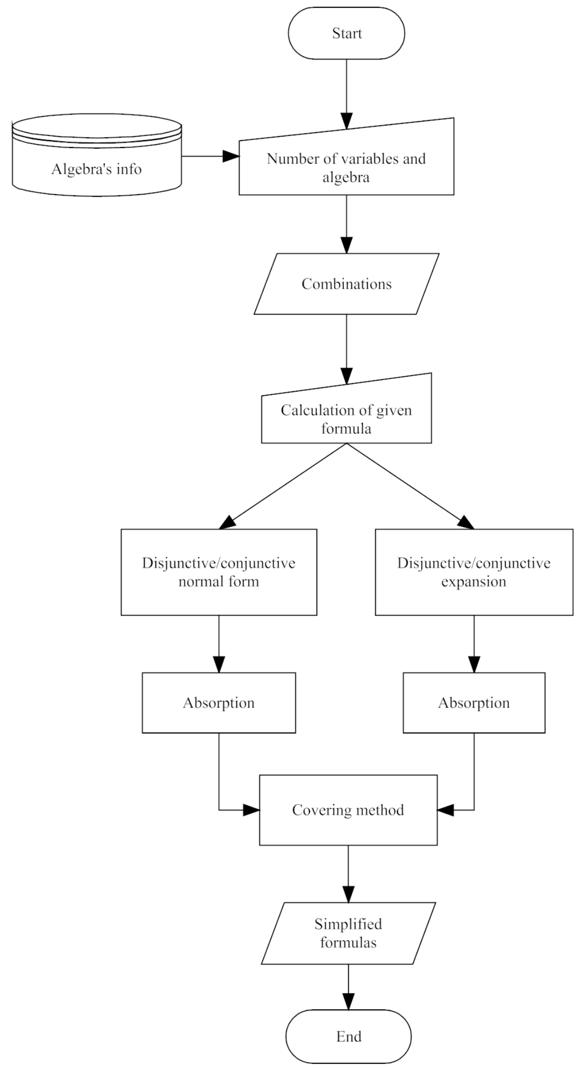



Figure 3. Simplification algorithm flowchart

intersections are identified between the elements in columns and rows which will determine the essential literals for the simplification and other literals, that although are not essential, have to be used to complete the simplification process. It is important to clarify that it is likely to get different solutions for the simplification, so the algorithm has to be able to found all of them and additionally to point out the solution that has fewer literals as the best one.

### A. The Normal Form

In order to find the normal form, first, the algebra and the number of variables have to be known to produce all of the possible combinations, and the information that describes the formula to simplify should be saved in an array; so elements are located into the array, and thus, a coincidence between this position and the combination position that corresponds

| | | Literal | | | |
|---|---|---|---|---|---|
| | | A | B | C | ... |
| Existence | ( ) | 1 | 0 | 1 | ... |
| | ( ') | 0 | 1 | 1 | ... |

Figure 4. Literal existence representation into an array

to each value is made. After that, the symbolic formula can be written by using the manipulation of literals explained in Section III-C.

### *B. The Disjunctive/Conjunctive Expansion*

The disjunctive/conjunctive expansion is found in a similar way to the normal form. However, this process involves algebraic operations among elements that describe the formula to simplify, and that implies operations among elements which belong to the same array. It is obtained that the disjunctive/conjunctive expansion will have as many elements as possible combinations including the omission of the variables.

The position of elements which should be operated into the array, the increment rate of the position in the array and the number of operations per increment, have a mathematical pattern that can be described based on the type of algebra and the number of variables previously known in the algorithm.

### *C. Literals Manipulation Techniques*

The manipulation of literals is very important during the whole simplification procedure, so it was appropriate to create a strategy to operate them and determine their logic values in an easy way, since types of algebra such as Kleene's and De Morgan's use values different from numbers. Furthermore, regarding the covering process, some values from the disjunctive/conjunctive expansion and the normal form have to be compared. To solve this problem, it was proposed to work with arrays where each position represents the existence of the literal. For instance, the Figure 4 shows the array that corresponds to $A{*}B'{*}C{*}C'{*}...$ where the literal $x$ has a position that is different from the position of $x'$, thus, it is possible to work with the numbers $1$ and $0$ indicating whether a literal exists or not in a certain expression.

## IV. Conclusion and Related Work

The presented algorithm in LabView® has allowed for the discovery of some simplifications for arbitrary formulas so that the equivalence could be verified in all of the possible values in the respective algebra. Therefore, it permits a first validation of the algorithm and the simplification method. However, the investigation continues in the process of metric definition and the performance. Currently, this method of simplification is being used in fuzzy control applications, so new design strategies are being studied and explored to propose them in this field and consequently analyze their differences with the methods already found in the literature.

## References


[1] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.

[2] G. J. Klir and B. Yuan, *Fuzzy sets and fuzzy logic: theory and applications*. New Jersey: Prentice Hall PTR, 1995.

[3] M. Gehrke, C. Walker, and E. Walker, "Normal forms and truth tables for interval-valued fuzzy logic," in *Joint 9th IFSA World Congress and 20th NAFIPS International Conference*, vol. 5, Vancouver, British Columbia, Canada, July 2001, pp. 1327–1331.

[4] ——, "Normal forms and truth tables for fuzzy logics," *Fuzzy Sets and Systems*, vol. 138, no. 1, pp. 25–51, August 2003.

[5] M. Mukaidono, "Algebraic structures of interval truth values in fuzzy logic," in *Proceedings of the Sixth IEEE International Conference On Fuzzy Systems*, vol. 3, Barcelona, Spain, July 1997, pp. 699–705.

[6] M. Gehrke, C. Walker, and E. Walker, "Some comments on interval valued fuzzy sets," *International Journal of Intelligent Systems*, vol. 11, no. 10, pp. 751–759, October 1996.

[7] G. D. Hachtel and F. Somenzi, *Logic Synthesis and Verification Algorithms*. Dordrecht: Kluwer Academic Publishers, 1996.

[8] A. Kandel, "On minimization of fuzzy functions," *IEEE Transactions On Computers*, vol. C-22, no. 9, pp. 826–832, September 1973.

[9] P. Siy and C. S. Chen, "Minimization of fuzzy functions," *IEEE Transactions On Computers*, vol. C-21, no. 1, pp. 100–102, January 1972.

[10] X. Zhiwei, "Multivalued logic and fuzzy logic - their relationship, minimization, and application to fault diagnosis," *IEEE Transactions On Computers*, vol. C-33, no. 7, pp. 679–681, July 1984.

[11] H. T. Nguyen and E. A. Walker, *A First Course in Fuzzy Logic*, 3rd ed. Boca Raton, Florida: Chapman & Hall/CRC, 2006.

[12] O. Salazar Morales, "Método de simplificación de fórmulas booleanas, borrosas cuasi-estándar y borrosas cuasi-estándar de intervalo por medio de álgebras finitas," Tesis de pregrado en Ingeniería Electrónica, Universidad Distrital Francisco José de Caldas. Facultad de Ingeniería, Bogotá D.C., Colombia, 2011.

[13] O. Salazar Morales and J. J. Soriano Méndez, "Quasi-standard fuzzy algebras. a short mathematical characterization," in *2011 IFSA World Congress and AFSS International Conference*, Surabaya and Bali Island, Indonesia, June 2011.