symmetry

Article

Generation of Polynomial Automorphisms Appropriate for the Generalization of Fuzzy Connectives

Eleftherios Makariadis [1,†], Stefanos Makariadis [2,†], Avrilia Konguetsof [2,†] and Basil Papadopoulos [2,*,†]

[1] Hellenic Air Force Academy, Department Supply Chain, Dekelia Air Base, 13677 Acharnes, Greece; elmakariadis@gmail.com
[2] Section of Mathematics, Programming and General Courses, Department of Civil Engineering, School of Engineering, Democritus University of Thrace, 67100 Xanthi, Greece; smakaria@civil.duth.gr (S.M.); akogkets@civil.duth.gr (A.K.)
* Correspondence: papadob@civil.duth.gr; Tel.: +30-2541079747
† These authors contributed equally to this work.

Abstract: Fuzzy logic is becoming one of the most-influential fields of modern mathematics with applications that impact not only other sciences, but society in general. This newly found interest in fuzzy logic is in part due to the crucial role it plays in the development of artificial intelligence. As a result, new tools and practices for the development of the above-mentioned field are in high demand. This is one of the issues this paper was composed to address. To be more specific, a sizable part of fuzzy logic is the study of fuzzy connectives. However, the current method used to generalize them is restricted to the use of basic automorphisms, which hinders the creation of new fuzzy connectives. For this reason, in this paper, a new method of generalization is conceived of that aims to generalize the fuzzy connectives using polynomial automorphism functions instead. The creation of these automorphisms is achieved through numerical analysis, an endeavor that is supported with programming applications that, using mathematical modeling, validate and visualize the research. Furthermore, the automorphisms satisfy all the necessary criteria that have been established for use in the generalization process and, consequently, are used to successfully generalize fuzzy connectives. The result of the new generalization method is the creation of new usable and flexible fuzzy connectives, which is very promising for the future development of the field.

Keywords: polynomial automorphisms; fuzzy logic; mathematical analysis; generalization of fuzzy connectives; fuzzy connectives; artificial intelligence; Lagrange interpolation; mathematical modeling

Citation: Makariadis, E.; Makariadis, S.; Konguetsof, A.; Papadopoulos, B. Generation of Polynomial Automorphisms Appropriate for the Generalization of Fuzzy Connectives. *Symmetry* **2023**, *15*, 1992. https://doi.org/10.3390/sym15111992

Academic Editors: Jian Zhou, Ke Wang, Yuanyuan Liu and José Carlos R. Alcantud

Received: 24 September 2023
Revised: 12 October 2023
Accepted: 25 October 2023
Published: 28 October 2023

1. Introduction

The main focus of this paper is the production of polynomial automorphisms, which are able to be used to generalize fuzzy connectives. Achieving this is important because of the crucial role that fuzzy connectives play in the fields of fuzzy logic and, consequently, artificial intelligence [1]. To be more specific, the above-mentioned fields will benefit from advancements in the study of fuzzy connectives, like the one presented in this paper, as they make possible further developments and optimizations. In order to further understand the significance of fuzzy connectives, the following paragraphs present the history of the field through the most-notable published research.

The study of fuzzy connectives started with the publication of the paper "Statistical metrics" by Menger [2], which used the concept of t-norms for the first time. Then, the field progressed with the work of Schweizer B.and Sklar A. [3] which spanned several years. Specifically, publications in 1958, 1960, 1961, and 1983 defined the theorems of ordinary rules and documented their findings. The work of Schweizer B. and Sklar A. was built upon by Ling C.H. [4], who, in 1965, determined the Archimedean t-norms. In 1979, Frank M.J. [5] expanded upon the previous research by defining the parameterized families of t-norms. However, the progress did not stop, as Navara M. [6] in 1999, Gottwald

S. [7] in 2000, and Klement E.P. [8] in 2001 proposed the strategy of generating t-norms via automorphism and additive generative functions. Modus ponens and tollens were discovered in 2004 by Kerre E. et al. [9]. In the same year, Kerre E. and Nachtegael M. [10] defined the fuzzy mathematical morphology. Four years later, in 2008, the implications of various properties were found by Trillas E. et al. [11].

However, previously, in 2006, Bustince H. et al. [12] published a work on fuzzy measures and image processing. Thereafter, Baczyński M. and Jayaram B., as well as Mas M. et al. [13,14], in 2007, crafted a method that generates (S,N)-implications. A strategy for creating R-implications was invented by Fodor J.C. and Roubens M. [15], in 1994. The generation of QL- and D-operations constituted the main objective of the process published by Mas M. et al. [16] in 2006. In 2004, the problem of generating f- and g-implications was tackled by Yager R.R. [17]. Finally, in 2012, Callejas C. et al. [18] proposed a method that creates any fuzzy implication.

For a decade, from 2012 to 2021, there was little to no further research about the subject of fuzzy connectives. Nonetheless, in 2022, Makariadis S. and Papadopoulos B. [19] published the paper titled "Generalization of Fuzzy Connectives", which brought the focus back to the field. This new development motivated the creation of this paper as a proof that the interest brought to the field by the 2022 paper was not temporary, but rather, the beginning to a series of new discoveries. So, the main goal of this work is to prove that the study of fuzzy connectives is not complete by developing a new method of generalizing fuzzy connectives. The conclusions drawn from the research presented in this paper are multiple. To be more specific, the field of the generalization of fuzzy connectives is proven to be still active, evidenced by the creation of a new method of generalizing fuzzy connectives that provides more flexibility and usability in comparison with established methods. Furthermore, it was concluded that other fields can be used in combination with fuzzy logic to create new tools and, finally, that automation paired with fuzzy logic can offer significant time savings gains. Finally, it is important to mention that the structure of the paper is as follows:

Introduction: In this section, a brief presentation of the research included in this paper takes place. Furthermore, the history, as well as the state of the paper's field are explored.

Preliminaries: In this section, the main concepts necessary for the understanding of the paper are defined.

Materials and Methods: In this section, the methods and strategies followed in the paper are explained in detail before being mathematically proven.

Results: In this section, the results of the paper are presented.

Discussion: In this section, the results of the paper are discussed and interpreted from the perspective of previous studies.

Conclusions: In this section, the conclusions drawn from the whole paper are presented.

2. Preliminaries

This section is dedicated to the display of the basic concepts used throughout the paper. To be more specific, the main points of interest are fuzzy negations, triangular norms (conjunctions and disjunctions), and fuzzy implications, as they form the basis of fuzzy connectives. Furthermore, emphasis is given to the explanation of notions such as automorphisms and Lagrange interpolation.

2.1. Fuzzy Negations

The definition displayed in this subsection can be found in the following references: Baczyński M., 1.4.1–1.4.2 Definitions, pp. 13–14, [20], Bedregal B.C., p. 1126, [21], Fodor J., 1.1–1.2 Definitions, p. 3, [15], Gottwald S., 5.2.1 Definition, p. 85, [7], Weber S., 3.1 Definition, p. 121, [22], and Trillas E., p. 49, [23].

Definition 1. *A function* $N : (0, 1) \to [0, 1]$ *is called a fuzzy negation if*
$(N1)$ $N(0) = 1$, $N(1) = 0$;
$(N2)$ *N is decreasing*.

A fuzzy negation N is called strict if, in addition to the former properties, the following apply:
$(N3)$ *N is strictly decreasing;*
$(N4)$ *N is continuous.*
A fuzzy negation N is called strong if the following property is satisfied:
$(N5)\ N(N(x)) = x,\ x \in [0,1]$.

2.2. Triangular Norms (Conjunctions)

In this subsection, the definition and properties of t-norms will be provided.

The sequent definition can be found in: Klement E.P et al., 1.1 Definition, pp. 4–10, [8], Baczyński M., 2.1.1, 2.1.2 Definitions, pp. 41–42, [20], Weber S., 2.1 Definition, pp. 116–117, [22], and Yun s., p. 16, [24].

Definition 2. *A function* $T : [0,1]^2 \to [0,1]$ *is called a triangular norm, or t-norm, if it satisfies, for all* $x, y \in [0,1]$*, the following conditions:*
$(T1)\quad T(x,y) = T(y,x), \quad (commutativity);$
$(T2)\quad T(x,T(y,z)) = T(T(x,y),z), \quad (associativity);$
$(T3)\quad if\ \ y \le z,\ \ then\ \ T(x,y) \le T(x,z), \quad (monotonicity);$
$(T4)\quad T(x,1) = x, \quad (boundary\ condition).$

2.3. Triangular Conorms (Disjunctions)

S-conorms allow for the generalization of the union in a lattice or disjunction in logic. The sequent definition can be found in: Klement E.P et al., 1.13 Definition, p. 11, [8], Baczyński M., 2.2.1, 2.2.2 Definitions, pp. 45–46, [20], and Yun s., p. 22, [24].

Definition 3. *A function* $S : [0,1]^2 \to [0,1]$ *is called a triangular conorm (t-conorm) if it satisfies, for all* $x, y \in [0,1]$*, the following conditions:*
$(S1) : S(x,\ y) = S(y,\ x)\ (commutativity);$
$(S2) : S(x,\ S(y,\ z)) = S(S(x,\ y),\ z)\ (associativity);$
$(S3)$: *If* $y \le z$, *then* $S(x,\ y) \le S(x,\ z)\ (monotonicity);$
$(S4) : S(x,\ 0) = x$ *(neutral element 0).*

2.4. Fuzzy Implications

Fuzzy implications play a crucial role in fuzzy logic as they represent what the classical implications are in crisp logic. The primary use of fuzzy implication functions is the execution of any fuzzy "if–then" rule on fuzzy systems.

The sequent definition can be found: Baczyński M., p. 2, [20], Yun s., p. 5, [24], and Fodor J., p. 299, [25].

Definition 4. *A binary operator* $I : [0,1]^2 \to [0,1]$ *is said to be an implication function, or an implication, if, for all* $x, y \in [0,1]$*, it satisfies:*
$(I1) : I(x,\ z) \ge I(y,\ z)\ \ when\ x \le y,\ \ isinthe\ first\ place\ (antitonicity);$
$(I2) : I(x,\ y) \le I(x,\ z)\ \ when\ y \le z,\ \ isinthe\ second\ place\ (isotonicity);$
$(I3) : I(0,0) = 1,\ (boundary\ condition);$
$(I4) : I(1,1) = 1,\ (boundary\ condition);$
$(I5) : I(1,0) = 0,\ (boundary\ condition).$

A function $I : [0,1]^2 \to [0,1]$ is called a fuzzy implication only if it satisfies $(I1)$–$(I5)$.

2.5. Automorphism Functions

Automorphisms are necessary for the generalization of fuzzy connectives, and as a result, they shape, to a degree, how the field evolves.

The sequent definition can be found in: Bedregal B., p. 1127, [21], and Yun s., p. 13, [24].

Definition 5. *A mapping* $\varphi : [a, b] \to [a, b]$ *(*$[a, b] \subset \mathbb{R}$*) is an automorphism of the interval [a, b] if it is continuous and strictly increasing and satisfies the boundary conditions:* $\varphi(a) = a$ *and* $\varphi(b) = b$*. If* φ *is an automorphism of the unit interval, then* φ^{-1} *is also an automorphism of the unit interval.*

2.6. Lagrange Interpolating Polynomial

The following definition was retrieved from [26].

Definition 6. *The Lagrange interpolating polynomial is the polynomial* $P(x)$ *of degree* $\leq (n-1)$ *that passes through the n points* $(x_1, y_1 = f(x_1)), (x_2, y_2 = f(x_2)), \ldots, (x_n, y_n = f(x_n))$*, and is given by*

$$P(x) = \sum_{j=1}^{n} P_j(x),$$

where

$$P_j(x) = y_j \prod_{\substack{k=1 \\ k \neq j}}^{n} \frac{x - x_k}{x_j - x_k}.$$

Written explicitly,

$$P(x) = \frac{(x-x_2)(x-x_3)\cdots(x-x_n)}{(x_1-x_2)(x_1-x_3)\cdots(x_1-x_n)}y_1 + \frac{(x-x_1)(x-x_3)\cdots(x-x_n)}{(x_2-x_1)(x_2-x_3)\cdots(x_2-x_n)}y_2 + \cdots + \frac{(x-x_1)(x-x_2)\cdots(x-x_{n-1})}{(x_n-x_1)(x_n-x_2)\cdots(x_n-x_{n-1})}y_n.$$

3. Materials and Methods

As mentioned in the above segments, this paper is focused on the creation of automorphisms that can be used to generalize fuzzy connectives. The generalization of fuzzy connectives using automorphisms was achieved with the use of one of the following equations:

$$N_\varphi(x) = \varphi^{-1}(N(\varphi(x))), \ \forall x \in [0,1] \tag{1}$$

$$T_\varphi(x,y) = \varphi^{-1}(T(\varphi(x), \varphi(y))), \ \forall x,y \in [0,1] \tag{2}$$

$$S_\varphi(x,y) = \varphi^{-1}(S(\varphi(x), \varphi(y))), \ \forall x,y \in [0,1] \tag{3}$$

$$I_\varphi(x,y) = \varphi^{-1}(I(\varphi(x), \varphi(y))), \ \forall x,y \in [0,1] \tag{4}$$

The process of generalization, as displayed above, (1)–(4), involves the use of an automorphism and a fuzzy connective. In [19], the approach of using different fuzzy connective families in conjunction with basic automorphisms was used to achieve generalization. This paper, however, explores the prospect of pairing well-known fuzzy connectives with polynomial automorphisms instead of basic ones.

The reason that moved the paper towards this direction is, firstly, that the fuzzy connective part of the equations has already been researched in regard to generalization and, secondly, the fact that the need for new automorphisms cannot be satisfied by the limited number of practical basic automorphisms (Figure 1). To be more specific, it is not a viable solution to expand further than a few degrees of the mentioned automorphisms, as doing so would lead to computationally intensive and complex scenarios. As a result, the need for new practical automorphisms remains unaddressed, and a task is accomplished in this paper by the use of polynomial automorphisms.

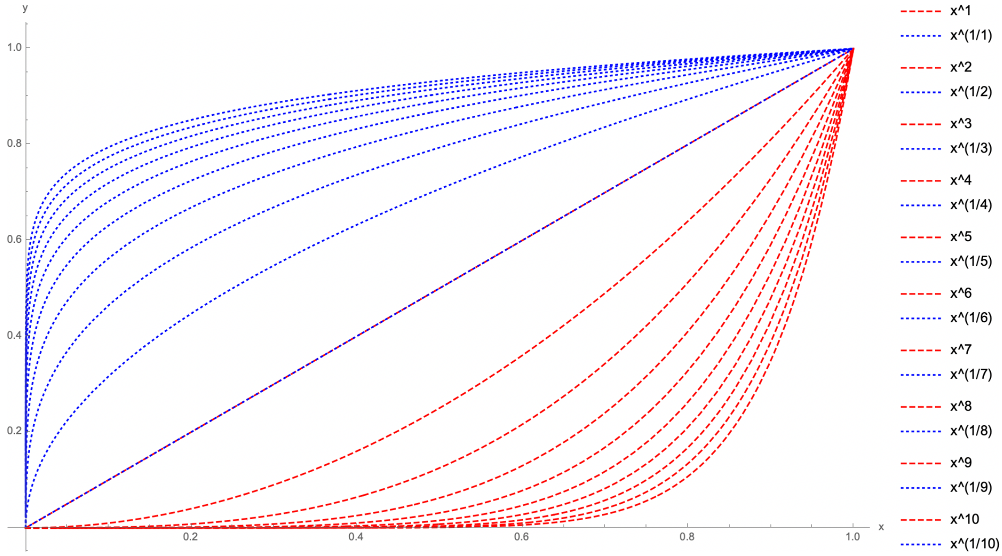

Figure 1. Graph of the basic automorphism functions.

However, currently, there are no appropriate automorphisms for use in the generalization of fuzzy connectives. So, before implementing the new strategy, it was necessary to craft a procedure for generating the necessary polynomial automorphisms. The main problem that was encountered by pursuing this endeavor is the fact that it is notoriously difficult to develop new automorphisms, with the task becoming even more complex when fuzzy logic is involved as the new polynomial functions must not only satisfy the automorphism criteria, but also the restrictions of fuzzy logic.

The problem was addressed by the composition of an algorithm that using numerical analysis methods generates the needed polynomial automorphisms. The algorithm is described in Theorem 1:

Theorem 1. *Three abscissas are chosen from* $[0,1]$*:* $x_0 = 0,\ x_1,\ 0 < x_1 < 1,\ x_2 = 1$.
Let $f : [0,1] \to [0,1]$ *be an automorphism function with the formula:* $f(x) = x \cdot \sqrt{x}$.
The corresponding ordinates are calculated with f*:* $f(0) = 0$, $f(x_1) = x_1 \cdot \sqrt{x_1}$, $f(1) = 1$.
The Lagrange interpolation formula:

$$L(x) = \frac{((x - x_1)(x - x_2))f(x_0)}{(x_0 - x_1)(x_0 - x_2)} + \frac{((x - x_0)(x - x_2))f(x_1)}{(x_1 - x_0)(x_1 - x_2)} + \frac{((x - x_0)(x - x_1))f(x_2)}{(x_2 - x_0)(x_2 - x_1)} \tag{5}$$

for the generation of second-degree polynomials is used.
The result is the generation of a second-degree polynomial with the formula:

$$L(x) = \frac{\sqrt{x_1} - 1}{x_1 - 1} \cdot x^2 + \frac{x_1 - \sqrt{x_1}}{x_1 - 1} \cdot x \tag{6}$$

Let $\varphi : [0,1] \to [0,1]$ *be a function with the formula:* $\varphi(x) = \frac{\sqrt{x_1}-1}{x_1-1} \cdot x^2 + \frac{x_1-\sqrt{x_1}}{x_1-1} \cdot x$.
Then, φ *is an invertible automorphism function.*

As is visible in Theorem 1, the algorithm responsible for the generation of the polynomial automorphisms has multiple steps. However, before explaining them, it was deemed necessary to highlight a key decision that shaped the final result.

The decision mentioned was about the degree of the generated polynomial. To be more specific, in this paper, the generation of second-degree polynomials was pursued because a major focus was given to the production of new automorphisms that can replace the basic ones while bypassing problems like complexity and computational resources, which plague higher degrees. However, it is important to note that, according to the results that the current strategy presents, the generation of higher-degree polynomials appropriate for fuzzy connective generalization is possible. This direction will be further explored in the "Discussion" Section of this paper.

The explanation of Theorem 1 begins with the choice of three abscissas. These values are utilized by the function f, which calculates the corresponding ordinates, leading to the creation of a set of three points. The choice of the function f is crucial to the success of this endeavor because it influences the properties of the produced polynomial. The main point of interest regarding those properties is the fact that the generated polynomials must satisfy the automorphism criteria for use in fuzzy logic. One of the criteria is the fact that the polynomial must cross the points (0,0) and (1,1). The result drawn from this realization is dual. Firstly, two of the three abscissas must be set to 0 and 1, so that the above-mentioned points can be realized. At this point, it is important to mention that the abscissa of the third point is a parameter with a value between 0 and 1 (as this is the interval to which fuzzy logic is applied). Secondly, a function f must be selected so that it can return the appropriate ordinates and complete the creation of the points. The crossing of these two points by the polynomial is guaranteed by the Lagrange interpolation, which produces polynomials that cross all the points that are given as an input and, consequently, crossing (0,0)–(1,1).

When researching functions that would be applicable for this use, the goal was to pinpoint a function that not only satisfied the above limitations, but had a power approximating the degree of the polynomial. In the beginning, the use of the x^2 function among other low-degree basic automorphisms was considered. However, the use of these functions was found to be problematic, as for particular values of the third abscissa, the polynomials they generated were basic automorphisms. As a result, the function $x \cdot \sqrt{x}$ was selected for the role of f. This decision was based on the fact that this function has a power that approximately is the nearest to a second-degree polynomial function while providing acceptable and consistent results.

Finally, the points are replaced to the Lagrange interpolation formula for second-degree polynomials, and a general automorphism is generated that includes the third abscissa value as a parameter. Furthermore, the generated automorphism is invertible, and as a result, it can be used for the generalization of fuzzy connectives. The innovative aspect of the automorphism-generation strategy presented in this paper is the fact that every value of the parameter can produce a unique invertible automorphism and, consequently, provide an infinite number of generalized fuzzy connectives. The proof of Theorem 1 is the following:

Proof of Theorem 1. Let $f : [0,1] \to [0,1]$ be an automorphism function with the formula: $f(\mathrm{x}) = x \cdot \sqrt{x}$.
Three abscissas are chosen from $[0,1]$: $x_0 = 0,\ x_1,\ 0 < x_1 < 1,\ x_2 = 1$.
The corresponding ordinates are calculated with f: $f(0)=0$, $f(x_1) = x_1 \cdot \sqrt{x_1}$, $f(1)=1$.
The points generated are replaced in the Lagrange interpolation (5), and the polynomial (6) is constructed:

$$L(x) = \frac{((x-x_1)(x-x_2))\cdot f(x_0)}{(x_0-x_1)(x_0-x_2)} + \frac{((x-x_0)(x-x_2))\cdot f(x_1)}{(x_1-x_0)(x_1-x_2)} + \frac{((x-x_0)(x-x_1))\cdot f(x_2)}{(x_2-x_0)(x_2-x_1)} \Leftrightarrow$$

$$L(x) = \frac{((x-x_1)(x-1))\cdot f(0)}{(0-x_1)(0-1)} + \frac{((x-0)(x-1))\cdot f(x_1)}{(x_1-0)(x_1-1)} + \frac{((x-0)(x-x_1))\cdot f(1)}{(1-0)(1-x_1)} \Leftrightarrow$$

$$L(x) = \frac{((x-x_1)(x-1))\cdot 0}{(0-x_1)(0-1)} + \frac{((x-0)(x-1))\cdot (x_1\cdot\sqrt{x_1})}{(x_1-0)(x_1-1)} + \frac{((x-0)(x-x_1))\cdot 1}{(1-0)(1-x_1)} \Leftrightarrow$$

$$L(x) = \frac{x(x-1)\cdot(x_1\cdot\sqrt{x_1})}{x_1\cdot(x_1-1)} - \frac{x(x-x_1)}{(x_1-1)} \Leftrightarrow$$

$$L(x) = \frac{\sqrt{x_1}-1}{x_1-1}\cdot x^2 + \frac{x_1-\sqrt{x_1}}{x_1-1}\cdot x$$

Let $\varphi : [0,1] \to [0,1]$ be a function with the formula:

$$\varphi(x) = \frac{\sqrt{x_1}-1}{x_1-1}\cdot x^2 + \frac{x_1-\sqrt{x_1}}{x_1-1}\cdot x \tag{7}$$

$$\varphi(0) = 0$$

$$\varphi(1) = 1$$

$$\varphi'(x) = \frac{2\cdot(\sqrt{x_1}-1)}{x_1-1}\cdot x + \frac{x_1-\sqrt{x_1}}{x_1-1}$$

$$x_1 < 1 \Leftrightarrow x_1 - 1 < 0$$

$$x_1 < 1 \Leftrightarrow \sqrt{x_1} < 1 \Leftrightarrow \sqrt{x_1} - 1 < 0$$

$$\frac{2\cdot(\sqrt{x_1}-1)}{x_1-1} > 0$$

$\varphi'(x) > 0 \Rightarrow \varphi$ strictly increasing $[0,1]$

$$\varphi(x) = y \Leftrightarrow \frac{\sqrt{x_1}-1}{x_1-1}\cdot x^2 + \frac{x_1-\sqrt{x_1}}{x_1-1}\cdot x = y \Leftrightarrow \frac{\sqrt{x_1}-1}{x_1-1}\cdot x^2 + \frac{x_1-\sqrt{x_1}}{x_1-1}\cdot x - y = 0$$

$$D = \left(\frac{x_1-\sqrt{x_1}}{x_1-1}\right)^2 - 4\cdot\left(\frac{\sqrt{x_1}-1}{x_1-1}\right)(-y) \Leftrightarrow$$

$$D = \frac{\left(x_1-\sqrt{x_1}\right)^2 + 4\cdot y\cdot(1-x_1)\cdot\left(1-\sqrt{x_1}\right)}{(1-x_1)^2}$$

$$x = \frac{x_1-\sqrt{x_1}+\sqrt{\left(x_1-\sqrt{x_1}\right)^2+4\cdot y\cdot(1-x_1)\cdot\left(1-\sqrt{x_1}\right)}}{2\cdot\left(1-\sqrt{x_1}\right)}$$

$$\varphi^{-1}(x) = \frac{x_1-\sqrt{x_1}+\sqrt{\left(x_1-\sqrt{x_1}\right)^2+4\cdot x\cdot(1-x_1)\cdot\left(1-\sqrt{x_1}\right)}}{2\cdot\left(1-\sqrt{x_1}\right)}$$

where D is the discriminant of the equation $\frac{\sqrt{x_1}-1}{x_1-1}\cdot x^2 + \frac{x_1-\sqrt{x_1}}{x_1-1}\cdot x - y = 0$.

So, the inverse of φ is:

$$\varphi^{-1}(x) = \frac{x_1-\sqrt{x_1}+\sqrt{(x_1-\sqrt{x_1})^2+4\cdot x\cdot(1-x_1)\cdot(1-\sqrt{x_1})}}{2\cdot(1-\sqrt{x_1})} \tag{8}$$

□

The graph of the generalized automorphism functions φ , φ^{-1} and their symmetry are shown in Figures 2–4, when a parameter $x_1 = 0.5$.

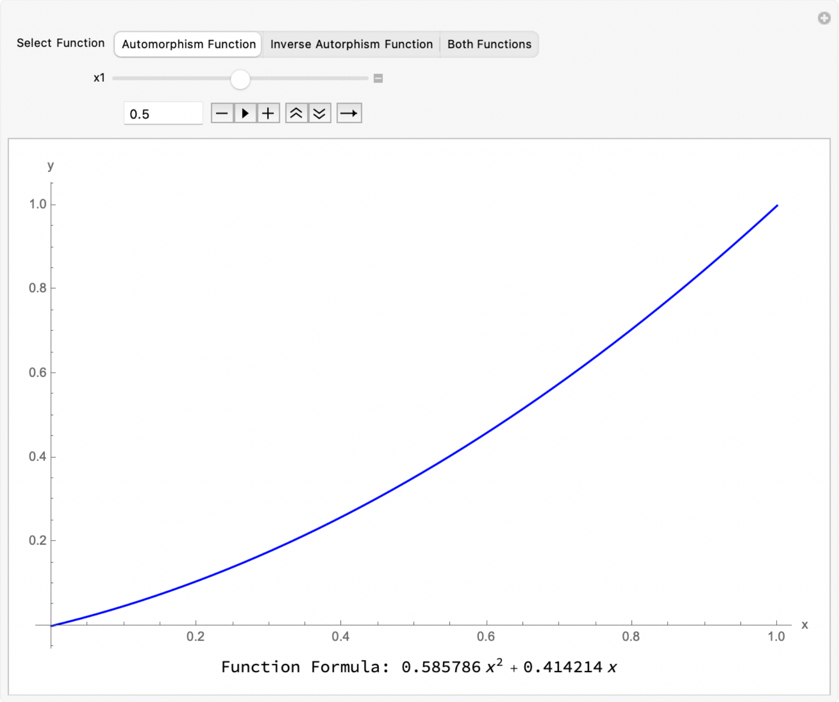

Figure 2. Graph of the polynomial automorphism function (when $x_1 = 0.5$ for illustration purposes).

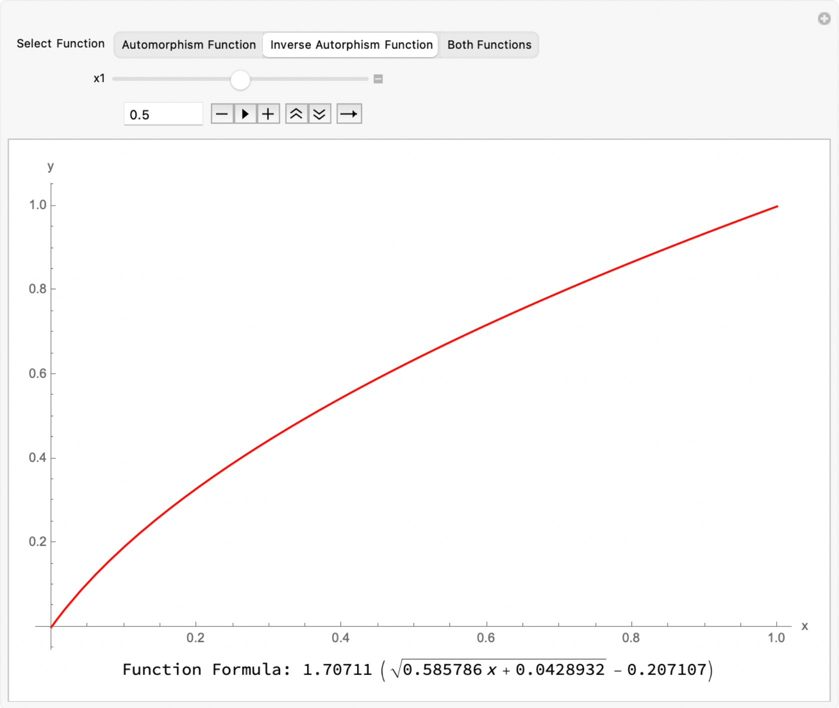

Figure 3. Graph of the inverse polynomial automorphism function (when $x_1 = 0.5$ for illustration purposes).

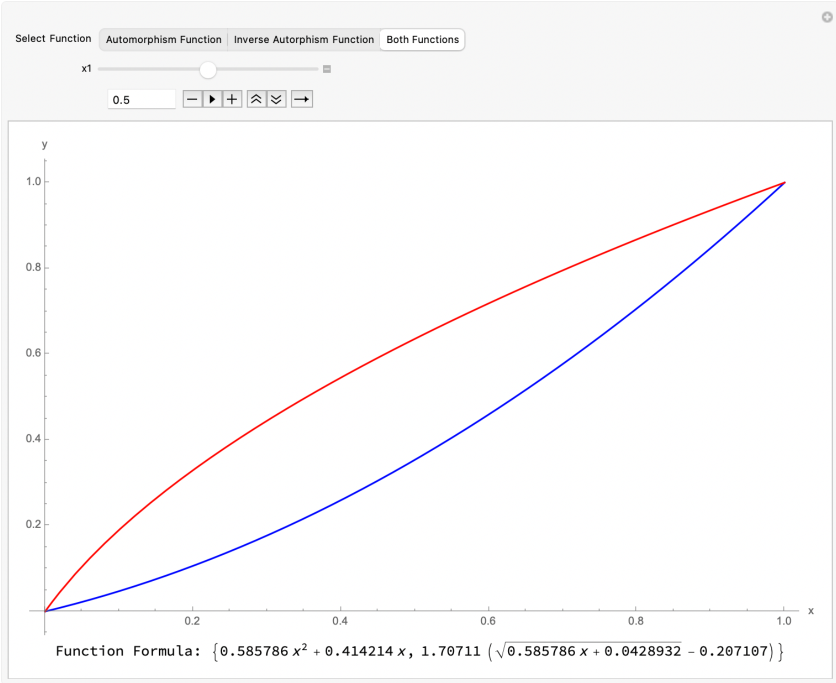

Figure 4. Graph displaying the symmetry of the polynomial automorphism and inverse polynomial automorphism functions (when $x_1 = 0.5$ for illustration purposes).

As mentioned at the start of this section, in this paper, the method of using polynomial automorphisms instead of basic ones for the generalization of fuzzy connectives is proposed. So, after proving the polynomial-automorphism-generation strategy, the main problem hindering the implementation of the new generalization method is resolved. Even though the new strategy generates polynomials that satisfy every criteria for use in the generalization of fuzzy connectives, it was deemed important to validate this statement. The simple use of the new polynomial automorphism in the generalization process would see the replacement of the parameter in Theorem 1 with a value and, then, the utilization of the result in one of (1)–(4). However, an example like this is too simple to present in this paper. As a result, the more-interesting Examples 1–4 were created. In each of these examples, a fuzzy connective generator is presented. To be more specific, each generator utilizes the polynomial automorphism general formula (with the parameter) in order to generalize an indicative fuzzy connective from each of the four categories of fuzzy connectives as the proof of the polynomial's effectiveness. The result of each generator is the generalized formula of the input fuzzy connective with a parameter that can take infinite values between 0 and 1 and, as a result, produce an infinite amount of generalized fuzzy connectives. The interesting aspect of these generators is that an infinite number of generalized fuzzy connectives can be generated from a single fuzzy connective input by simply replacing the parameter's value.

In Example 1, a natural negation is used as the input fuzzy connective during the generalization process. It is important to note the fact that the natural negation can be replaced only by other strong negations, and the current choice serves only the purpose of presenting the capabilities of the proposed direction.

Example 1. *Let $\varphi : [0,1] \to [0,1]$ be a function with Formula (7).*
Let $\varphi^{-1} : [0,1] \to [0,1]$ be the inverse function of φ with Formula (8).
Let $N : [0,1] \to [0,1]$ be a strong (natural) negation $N(x) = 1 - x$.
Then, there is a function $N_\varphi : \ [0,1] \to [0,1]$ with the formula:

$$N_\varphi(x) = \frac{x_1 - \sqrt{x_1} + \sqrt{(x_1 - \sqrt{x_1})^2 + 4 \cdot (1 - \frac{\sqrt{x_1}-1}{x_1-1} \cdot x^2 - \frac{x_1-\sqrt{x_1}}{x_1-1} \cdot x).(1 - x_1) \cdot (1 - \sqrt{x_1})}}{2 \cdot (1 - \sqrt{x_1})} \quad (9)$$

where x_1 is a parameter with values ranging in the interval $(0,1)$.

Indeed,
utilizing Theorem 1 of [19], *the formula* $N_\varphi(x) = \varphi^{-1}(N(\varphi(x)))$ *is applied.*
Then, Equations (7) and (8), *as well as the (natural) negation are replaced in the formula:*

$$N_\varphi(x) = \varphi^{-1}(N(\varphi(x))) \Leftrightarrow N_\varphi(x) = \frac{x_1-\sqrt{x_1}+\sqrt{\left(x_1-\sqrt{x_1}\right)^2+4\cdot N(\varphi(x)).(1-x_1)\cdot\left(1-\sqrt{x_1}\right)}}{2\cdot\left(1-\sqrt{x_1}\right)} \Leftrightarrow$$

$$N_\varphi(x) = \frac{x_1-\sqrt{x_1}+\sqrt{\left(x_1-\sqrt{x_1}\right)^2+4\cdot(1-\varphi(x)).(1-x_1)\cdot\left(1-\sqrt{x_1}\right)}}{2\cdot\left(1-\sqrt{x_1}\right)} \Leftrightarrow$$

$$N_\varphi(x) = \frac{x_1-\sqrt{x_1}+\sqrt{\left(x_1-\sqrt{x_1}\right)^2+4\cdot(1-\frac{\sqrt{x_1}-1}{x_1-1}\cdot x^2-\frac{x_1-\sqrt{x_1}}{x_1-1}\cdot x).(1-x_1)\cdot\left(1-\sqrt{x_1}\right)}}{2\cdot\left(1-\sqrt{x_1}\right)}$$

where x_1 is a parameter with values ranging in the interval $(0,1)$.

The graph of the generalized strong negation N_φ when a parameter $x_1 = 0.5$ shown in the Figure 5.

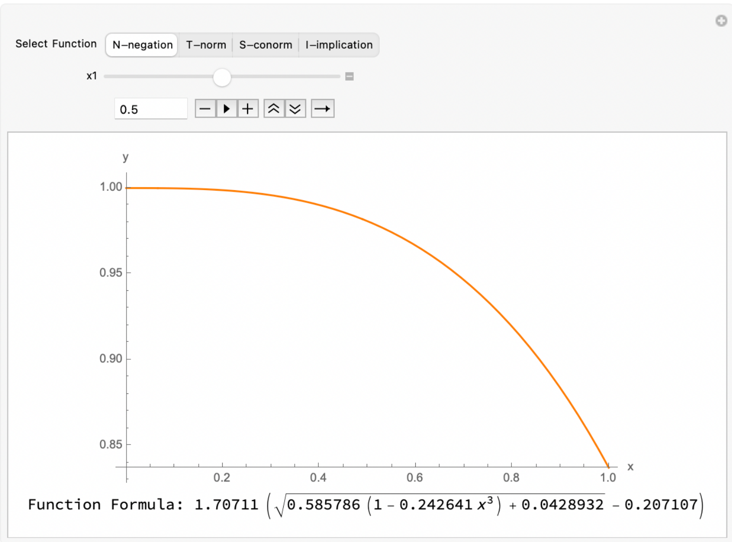

Figure 5. Graph of the generalized strong N negation (when $x_1 = 0.5$ for illustration purposes).

In Example 2, the minimum t-norm is used as the input fuzzy connective. In this generator configuration, the minimum t-norm can be replaced only by other t-norms.

Example 2. *Let* $\varphi : [0,1] \to [0,1]$ *be a function with Formula* (7).
Let $\varphi^{-1} : [0,1] \to [0,1]$ *be the inverse function of* φ *with Formula* (8).
Let $T : [0,1]^2 \to [0,1]$ *be a t-norm with the formula:* $T(x,y) = min\{x,y\}$.
Then, there is a function $T\varphi : [0,1]^2 \to [0,1]$ *with the formula:*

$$T_\varphi(x,y) = \frac{x_1-\sqrt{x_1}+\sqrt{(x_1-\sqrt{x_1})^2+4\cdot(\min\{\frac{\sqrt{x_1}-1}{x_1-1}\cdot x^2+\frac{x_1-\sqrt{x_1}}{x_1-1}\cdot x,\ \frac{\sqrt{x_1}-1}{x_1-1}\cdot y^2+\frac{x_1-\sqrt{x_1}}{x_1-1}\cdot y\})\cdot(1-x_1)\cdot(1-\sqrt{x_1})}}{2\cdot(1-\sqrt{x_1})} \quad (10)$$

where x_1 is a parameter with values ranging in the interval $(0,1)$.

Indeed,
utilizing Theorem 2 of [19], *the formula* $T_\varphi(x,y) = \varphi^{-1}(T(\varphi(x),\varphi(y)))$ *is applied.*
Then, Equations (7) *and* (8), *as well as the t-minimum-norm are replaced in the formula:*

$$T_\varphi(x,y) = \varphi^{-1}(T(\varphi(x),\varphi(y))) \Leftrightarrow T_\varphi(x,y) = \frac{x_1-\sqrt{x_1}+\sqrt{(x_1-\sqrt{x_1})^2+4\cdot(T(\varphi(x),\varphi(y)))\cdot(1-x_1)\cdot(1-\sqrt{x_1})}}{2\cdot(1-\sqrt{x_1})} \Leftrightarrow$$

$$T_\varphi(x,y) = \frac{x_1-\sqrt{x_1}+\sqrt{(x_1-\sqrt{x_1})^2+4\cdot(\min\{\varphi(x),\varphi(y)\})\cdot(1-x_1)\cdot(1-\sqrt{x_1})}}{2\cdot(1-\sqrt{x_1})} \Leftrightarrow$$

$$T_\varphi(x,y) = \frac{x_1-\sqrt{x_1}+\sqrt{(x_1-\sqrt{x_1})^2+4\cdot(\min\{\frac{\sqrt{x_1}-1}{x_1-1}\cdot x^2+\frac{x_1-\sqrt{x_1}}{x_1-1}\cdot x, \frac{\sqrt{x_1}-1}{x_1-1}\cdot y^2+\frac{x_1-\sqrt{x_1}}{x_1-1}\cdot y\})\cdot(1-x_1)\cdot(1-\sqrt{x_1})}}{2\cdot(1-\sqrt{x_1})}$$

where x_1 is a parameter with values ranging in the interval $(0,1)$.

The graph of the generalized t-norm T_φ when a parameter $x_1 = 0.5$ shown in the Figure 6.

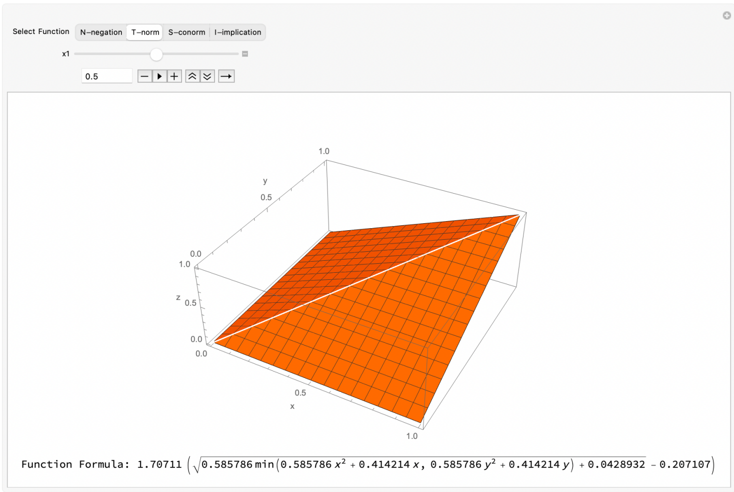

Figure 6. Graph of the generalized t-norm (when $x_1 = 0.5$ for illustration purposes).

In Example 3, the maximum S-conorm is used as the input fuzzy connective. In this generator configuration, the maximum S-conorm can be replaced only by other S-conorms.

Example 3. *Let $\varphi : [0,1] \to [0,1]$ be a function with Formula (7).*
Let $\varphi^{-1} : [0,1] \to [0,1]$ be the inverse function of φ with Formula (8).
Let $S : [0,1]^2 \to [0,1]$ be a S-conorm with the formula: $S(x,y) = max\{x,y\}$.
Then, there is a function $S\varphi : [0,1]^2 \to [0,1]$ with the formula:

$$S_\varphi(x,y) = \frac{x_1 - \sqrt{x_1} + \sqrt{(x_1 - \sqrt{x_1})^2 + 4\cdot(\max\{\frac{\sqrt{x_1}-1}{x_1-1}\cdot x^2 + \frac{x_1-\sqrt{x_1}}{x_1-1}\cdot x, \frac{\sqrt{x_1}-1}{x_1-1}\cdot y^2 + \frac{x_1-\sqrt{x_1}}{x_1-1}\cdot y\})\cdot(1-x_1)\cdot(1-\sqrt{x_1})}}{2\cdot(1-\sqrt{x_1})} \tag{11}$$

where x_1 is a parameter with values ranging in the interval $(0,1)$.

Indeed,
utilizing Theorem 3 of [19], the formula $S_\varphi(x,y) = \varphi^{-1}(S(\varphi(x),\varphi(y)))$ is applied.
Then, Equations (7) and *(8) as well as the S-conorm are replaced in the formula:*

$$S_\varphi(x,y) = \varphi^{-1}(S(\varphi(x),\varphi(y))) \Leftrightarrow S_\varphi(x,y) = \frac{x_1-\sqrt{x_1}+\sqrt{(x_1-\sqrt{x_1})^2+4\cdot(S(\varphi(x),\varphi(y)))\cdot(1-x_1)\cdot(1-\sqrt{x_1})}}{2\cdot(1-\sqrt{x_1})} \Leftrightarrow$$

$$S_\varphi(x,y) = \frac{x_1-\sqrt{x_1}+\sqrt{(x_1-\sqrt{x_1})^2+4\cdot(\max\{\varphi(x),\varphi(y)\})\cdot(1-x_1)\cdot(1-\sqrt{x_1})}}{2\cdot(1-\sqrt{x_1})} \Leftrightarrow$$

$$S_\varphi(x,y) = \frac{x_1-\sqrt{x_1}+\sqrt{(x_1-\sqrt{x_1})^2+4\cdot(\max\{\frac{\sqrt{x_1}-1}{x_1-1}\cdot x^2+\frac{x_1-\sqrt{x_1}}{x_1-1}\cdot x, \frac{\sqrt{x_1}-1}{x_1-1}\cdot y^2+\frac{x_1-\sqrt{x_1}}{x_1-1}\cdot y\})\cdot(1-x_1)\cdot(1-\sqrt{x_1})}}{2\cdot(1-\sqrt{x_1})}$$

where x_1 is a parameter with values ranging in the interval $(0,1)$.

The graph of the generalized s-conorm S_φ when a parameter $x_1 = 0.5$ shown in the Figure 7.

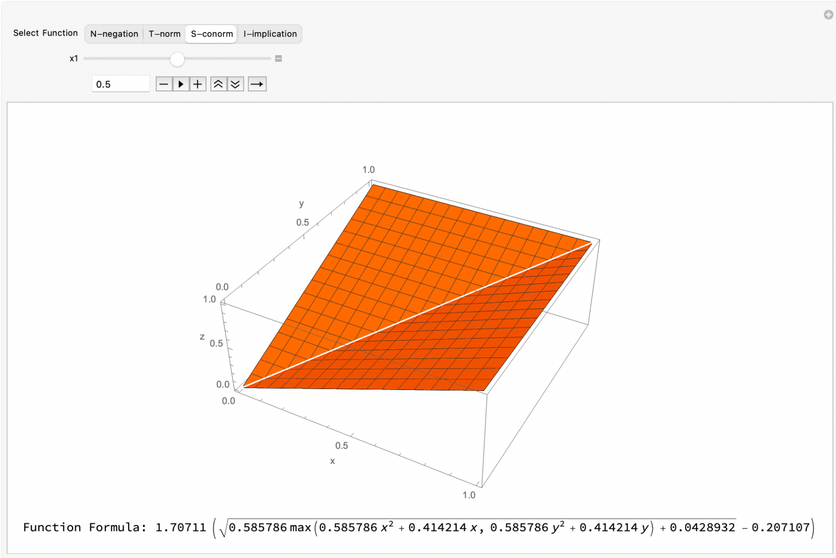

Figure 7. Graph of the generalized S-conorm (when $x_1 = 0.5$ for illustration purposes).

In Example 4, the Łukasiewicz I-implication is used as the input fuzzy connective. In this generator configuration, the Łukasiewicz I-implication can be replaced only by other I-implications.

Example 4. *Let $\varphi : [0,1] \to [0,1]$ be a function with Formula* (7).
Let $\varphi^{-1} : [0,1] \to [0,1]$ be the inverse function of φ with Formula (8).
Let $I : [0,1]^2 \to [0,1]$ be a Łukasiewicz I-implication with formula: $I(x,y) = min\{1, 1 - x + y\}$.
Then, there is a function $I_\varphi : \ [0,1]^2 \to [0,1]$ with the formula:

$$I_\varphi(x,y) = \frac{x_1 - \sqrt{x_1} + \sqrt{(x_1 - \sqrt{x_1})^2 + 4 \cdot (\min\{1, 1 - \frac{\sqrt{x_1}-1}{x_1-1} \cdot x^2 - \frac{x_1-\sqrt{x_1}}{x_1-1} \cdot x + \frac{\sqrt{x_1}-1}{x_1-1} \cdot y^2 + \frac{x_1-\sqrt{x_1}}{x_1-1} \cdot y\}) \cdot (1 - x_1) \cdot (1 - \sqrt{x_1})}}{2 \cdot (1 - \sqrt{x_1})} \quad (12)$$

where x_1 is a parameter with values ranging in the interval $(0,1)$.

Indeed,
utilizing [27], *the formula $I_\varphi(x,y) = \varphi^{-1}(I(\varphi(x), \varphi(y)))$ is applied.*
Then, Equations (7) and (8), *as well as the I-implication are replaced in the formula:*

$$I_\varphi(x,y) = \varphi^{-1}(I(\varphi(x), \varphi(y))) \Leftrightarrow I_\varphi(x,y) = \frac{x_1 - \sqrt{x_1} + \sqrt{(x_1-\sqrt{x_1})^2 + 4 \cdot (I(\varphi(x), \varphi(y))) \cdot (1-x_1) \cdot (1-\sqrt{x_1})}}{2 \cdot (1-\sqrt{x_1})} \Leftrightarrow$$

$$I_\varphi(x,y) = \frac{x_1 - \sqrt{x_1} + \sqrt{(x_1-\sqrt{x_1})^2 + 4 \cdot (\min\{1, 1-\varphi(x)+\varphi(y)\}) \cdot (1-x_1) \cdot (1-\sqrt{x_1})}}{2 \cdot (1-\sqrt{x_1})} \Leftrightarrow$$

$$I_\varphi(x,y) = \frac{x_1 - \sqrt{x_1} + \sqrt{(x_1-\sqrt{x_1})^2 + 4 \cdot (\min\{1, 1 - \frac{\sqrt{x_1}-1}{x_1-1} \cdot x^2 - \frac{x_1-\sqrt{x_1}}{x_1-1} \cdot x + \frac{\sqrt{x_1}-1}{x_1-1} \cdot y^2 + \frac{x_1-\sqrt{x_1}}{x_1-1} \cdot y\}) \cdot (1-x_1) \cdot (1-\sqrt{x_1})}}{2 \cdot (1-\sqrt{x_1})}$$

where x_1 is a parameter with values ranging in the interval $(0,1)$.

The graph of the generalized I-implication I_φ when a parameter $x_1 = 0.5$ shown in the Figure 8.

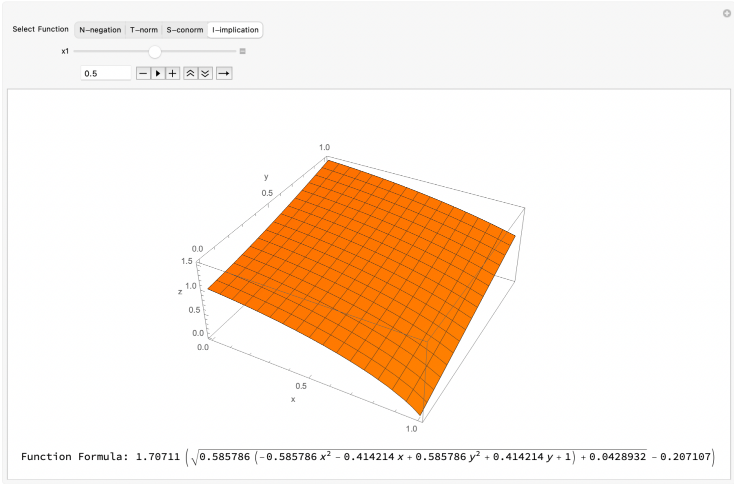

Figure 8. Graph of the generalized I-implication (when $x_1 = 0.5$ for illustration purposes).

As it is apparent from the above examples, the strategy of using polynomial automorphisms for the generalization of fuzzy connectives is a multistage and technical procedure. So, in this paper, mathematical modeling was employed for the purpose of automating the theorems as an efficient, reliable, and practical tool that streamlines the whole process from start to finish. In order to achieve this goal, two different paths during the development of the tool were taken. To be more specific: the tool was coded in two different programming languages and philosophies, which, even though they provide the same end results, prioritize different aspects.

The first version of the tool was coded in the programming language "MATLAB" using the version 2021b of the product "MATLAB", which is distributed by the company MathWorks. Before explaining the mentioned code, it is important to state that it has been saved in a public repository and is accessible at Supplementary Materials.

The complexity and size of the code may be a problem for researchers that are not familiar with computer programming. As a result, it was deemed necessary to provide a simpler and more-accessible version of the project written in pseudo code.

The pseudo code version of the "MATLAB" script in steps is given as follows:

- Step 1: Create the points that will be used in the Lagrange interpolation as X and Y vectors;
- Step 2: Implement the Lagrange interpolation algorithm;
- Step 3: Display the messages to the user regarding the results of Step 2;
- Step 4: Create and display the Lagrange interpolating polynomial;
- Step 5: Display the automorphism properties to the user for the verification of its validity;
- Step 6: Create and display of the inverse Lagrange interpolating polynomial;
- Step 7: Plot the polynomial automorphism and its inverse;
- Step 8: Display the generalization menu to the user;
- Step 9: Generalize and plot the chosen fuzzy connective.

Since the basic structure of the script has been presented, in the following paragraphs, a more-detailed explanation of the procedures and user interface used is give. Moreover, the instructions for the proper use and modification of the code are given so other researchers can maximize the utility of the research presented in this paper.

The code has a very simple, yet practical user interface. Firstly, upon running the program, the user will be requested to input a float number between the numbers 0 and 1. If the input code is not valid, the program will not continue and the process will be repeated until the user provides the expected values. When this happens, the code automatically starts the implementation of Theorem 1 by creating the set of points that will be used in the Lagrange interpolation. Then, the Lagrange interpolation is executed. The code snippet responsible for the code implementation of the Lagrange interpolation was retrieved from [28]; however, it has been modified to fit the goals and practices of this study. The output of the interpolation is a vector, which includes the coefficients of the generated second-degree polynomial. This vector is displayed to the user among the abscissas and ordinates of the points used in the Lagrange interpolation. Then, using the tools of the MATLAB environment, the vector is converted into a symbolic function that houses the automorphism function. Afterwards, the automorphism properties of the polynomial are displayed, which include the calculation and plotting of the derivative of the polynomial. Furthermore, the inverse of the polynomial automorphism is calculated, and it is plotted with the polynomial.

At this point, the user is met with a menu of options regarding the generalization of fuzzy connectives. Specifically, the user has to choose between the generalization of one of the four categories of fuzzy connectives or the termination of the program. As before, any invalid input of the user will lead to a repeat of the process. If one of the four categories of fuzzy connectives is chosen, the program creates an indicative fuzzy connective as in Examples 1, 2, 3, and 4 and stores it in a symbolic function. It is important to highlight that the fuzzy connectives that are used can be replaced by any other compatible fuzzy connective, with the choice presented in this paper being made for illustration purposes. Any researcher that is interested in using the tool can easily change the input fuzzy connective from the source code (the lines that the user can modify are pointed out with comments inside the code, e.g., line 83). Then, the appropriate formula for the generalization of fuzzy connectives via automorphisms is used in combination with the generated polynomial automorphism, its inverse, and the fuzzy connective. The generated connective is displayed and plotted either as a regular plot or as a 3D plot depending the fuzzy connective category.

The main philosophy of this approach is the fact that the code simplifies the process of the generalization of fuzzy connectives and guides the user throughout it. To be more specific, the user will generate a specific automorphism and, gradually, without any effort, will be provided with all the necessary information regarding the specific generalization that he/she intends to realize.

The second version of the tool was coded in the programming language "Wolfram Language" using the version 12 of the product "WOLFRAM MATHEMATICA", which is distributed by the company "WOLFRAM". The code is also available at Supplementary Materials. The main philosophy of the approach taken in this code is the visualization of the research presented in this paper and not the creation of a streamlined process. The motivation behind the development of this project was the creation of a tool capable of providing a visual representation of the paper's research that can be used and understood by any reader regardless of background. In order to achieve this, the program was divided into two independent sections, one focused on the presentation of the generated polynomial automorphisms and the other on the presentation of the generalized connectives.

In the first section, the code utilizes the general formulas (7 and 8, respectively) of the polynomial automorphism and its inverse as they were defined by Theorem 1 in order to create the respective plots. However, the program does not simply display the plots of these functions, as a more-elaborate user interface has been created. To be more specific, the plots have been labeled with the functions they represent and categorized into an easy-to-understand menu. Furthermore, each graph has its own unique color and a separate box that displays the specific function formula generated with every x_1 value. x_1 is defined by the user with the help of a slider, which simplifies the process even more. Moreover, a set of controls was implemented in the code that give the user the ability to influence the x_1 slider in multiple ways. For example, the user may choose to replace automatically different x_1 values in order to see the deviations between the unique generated polynomials. Finally, the menu presents the option of plotting the two functions on the same axis with all the above options being available.

In the second section, the code utilizes the generalized parametric Formulas ((9), (10), (11), and (12), respectively) as defined by Examples 1–4 in order to create the respective plots. The generalized parametric formulas used can be replaced by the generalized parametric formulas of any other compatible fuzzy connective, with the choice presented in this paper being made for illustration purposes. If a researcher wants to utilize the tool in order to visualize his/her own generalized fuzzy connective, he/she can follow the steps presented in the above-mentioned examples with the fuzzy connective of his/her choice in order to create the appropriate generalized fuzzy formula. The user interface and the various features of the program are the same as the above code, with the difference being that the majority of the plots are 3D and, as a result, were graphed in a different environment using the same color for visualization purposes.Finally, a third Mathematica script was created that was used to generate Figure 1, which displays the basic automorphism functions.

4. Results

The goal set at the start of the paper was the creation of a new method for generalizing fuzzy connectives using polynomial automorphisms. However, in order to achieve this target, it was necessary to generate the polynomial automorphisms, a task that was fulfilled with the composition of a strategy that involved numerical analysis. The results drawn from this research are represented in the following paragraphs.

4.1. Generation of Polynomial Automorphisms

The generation of polynomial automorphisms that satisfy the special criteria for use in fuzzy logic has proven to be a very demanding endeavor. However, the process displayed in this paper proved to be reliable and robust, as it is capable of generating an infinite number of unique polynomial automorphisms. Furthermore, two computer programs were constructed that automate the above-mentioned strategy into a procedure that can be operated by a researcher of any field. So, it is safe to say that the attempt to create a polynomial automorphism generator was successful.

4.2. Generalization of Fuzzy Connectives

Since the required automorphisms were successfully generated, the next step during the development process of this paper was the generalization of fuzzy connectives. In order to establish that the proposed method of generalization is valid, an indicative fuzzy connective was generalized from each fuzzy connective category. It is important to note at this point that the method can generalize any fuzzy connective; however, it is not possible to display the generalization of every fuzzy connective in a paper. The results of every generalization process were positive. As a result, the method presented in this paper was confirmed to work with any category of fuzzy connectives. Finally, the option of automating the generalization of the fuzzy connectives was added to the above-mentioned programs, leading to the creation of an efficient implementation of the introduced generalization method.

The main results of the paper can be briefly described as follows: a new powerful and robust polynomial automorphism generator was created; two programs were coded that automate both the generation of the polynomial automorphisms and the generalization of fuzzy connectives; a new, more-flexible method for generalizing fuzzy connectives was composed and proven.

5. Discussion

The results of the research presented in this paper provide a fresh view of the study of fuzzy connectives. One of the reasons that contributed to this is the fact that the generalization method proposed innovates in comparison with previous studies. To be more specific, every previous attempt at generalizing fuzzy connectives had turned to the use of alternative fuzzy connectives as a generalizing element. However, in this paper, the prospect of using a polynomial automorphism instead of a basic one was explored and validated, a choice that differentiated the method from any other that preceded it. Another reason is the fact that computer programming was employed to reinforce and automate the findings of the paper, a move that introduced new capabilities to the study of fuzzy connectives like the easy and accurate visualization of research. Furthermore, the discovery made in this paper opens the possibility of using numerical analysis as a potential tool in the study of fuzzy logic. Moreover, it is important to point out the fact that the generalized fuzzy connectives that were created by this paper are a very powerful and useful asset to the researchers that are active in fields like decision-making, artificial intelligence, and even environmental study like [29,30]. It is important also to mention that the use of second-degree polynomials in the process of generalization saved valuable processing resources and significantly reduced the complexity of the whole procedure, helping its widespread adoption. Finally, the utilization of degree polynomials highlights the future research direction of generating polynomial automorphisms of various powers.

6. Conclusions

This paper successfully tackled the problems and failings that led to its creation. Furthermore, it delivered useful and practical results that enable interesting future research directions. Furthermore, the presented research proved that computer programming can assist in the development of fuzzy connectives and fuzzy logic in general by automating tedious and repetitive tasks. In conclusion, taking everything this paper has researched into consideration, it is safe to assume that the study of fuzzy connectives not only is not near completion, but is at its infancy with exciting new opportunities becoming a reality.

Supplementary Materials: The supplementary materials can be available at: https://github.com/Eleftherios-Makariadis/Generation_of_Polynomial_Automorphisms_Appropriate_for_the_Generalization_of_Fuzzy_Connectives.

Author Contributions: Conceptualization, S.M.; methodology, S.M. and E.M.; software, E.M.; validation, E.M., S.M., A.K. and B.P.; investigation, E.M.; writing—original draft preparation, E.M.; writing—review and editing, E.M. and A.K.; visualization, E.M.; supervision, B.P.; project administration, B.P. and A.K.; funding acquisition, A.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: No data available, because the present paper does not use data.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Klement, E.P.; Slany, W. *Fuzzy Logic in Artificial Intelligence*; Marcel Dekker: Vienna, Austria, 1997; pp. 179–190.
2. Menger, K. A survey on fuzzy implication functions. *Proc. Natl. Acad. Sci. USA* **1942**, *28*, 535–537. [CrossRef] [PubMed]
3. Schweizer, B.; Sklar, A. *Probabilistic Metric Spaces*; Dover Publications, Inc.: Mineola, NY, USA, 1983.
4. Ling, C. Representation of associative functions. *Publ. Math. Debrecen* **1965**, *12*, 189–212. [CrossRef]
5. Frank, M. On the simultaneous associativity of f(x,y), and x+y-f(x,y). *Aequationes Math.* **1979**, *19*, 194–226. [CrossRef]
6. Navara, M. Characterization of Measures Based on Strict Triangular Norms. *J. Math. Anal. Appl.* **1999**, *236*, 370–383. [CrossRef]
7. Gottwald, S. *A Treatise on Many-Valued Logics*; Research Studies Press: Baldock, UK, 2001; pp. 63–105.
8. Klement, E.; Mesiar, R.; Pap, E. *Triangular Norms*; Kluwer: Dordrecht, The Netherlands, 2000; pp. 4–10, 108–110.
9. Kerre, E.; Huang, C.; Ruan, D. *Fuzzy Set Theory and Approximate Reasoning*; Wu Han University Press: Wuhan, China, 2004.
10. Kerre, E.; Nachtegael, M. *Fuzzy Techniques in Image Processing*; Springer: Berlin/Heidelberg, Germany, 2000; Volume 52.
11. Trillas, E.; Mas, M.; Monserrat, M.; Torrens, J. On the representation of fuzzy rules. *Int. J. Approx. Reason.* **2008**, *48*, 583–597. [CrossRef]
12. Bustince, H.; Mohedano, V.; Barrenechea, E.; Pagola, M. Definition and construction of fuzzy DI-subsethood measures. *Inf. Sci.* **2006**, *176*, 3190–3231. [CrossRef]
13. Baczyński, M.; Jayaram, B. On the characterizations of implications. *Fuzzy Sets and Systems* **2007**, *158*, 1713–1727. [CrossRef]
14. Mas, M.; Monserrat, M.; Torrens, J.; Trillas, E. A survey on fuzzy implication functions. *Proc. IEEE Trans. Fuzzy Syst.* **2007**, *15*, 1107–1121. [CrossRef]
15. Fodor, J.; Roubens, M. *Fuzzy Preference Modelling and Multicriteria Decision Support*; Kluwer Academic Publishers: Dordrecht, The Netherlands, 1994; Volume D, pp. 3–16.

16. Mas, M.; Monserrat, M.; Torrens, J. QL versus D-implications. *Kybernetika* **2006**, *42*, 351–366.
17. Yager, R. On some new classes of implication operators and their role in approximate reasoning. *Inf. Sci.* **2004**, *167*, 193–216. [CrossRef]
18. Callejas, C.; Marcos, J.; Bedregal, B. Actions of Automorphisms on Some Classes of Fuzzy Bi-implications. *SBMAC* **2012**, *177*, 140–146.
19. Makariadis, S.; Papadopoulos, B. Generalization of Fuzzy Connectives. *Axioms* **2022**, *11*, 130. [CrossRef]
20. Baczyński, M.; Jayaram, B. *Fuzzy Implications*; Springer: Berlin/Heidelberg, Germany, 2008; Volume 231.
21. Bedregal, B. On Fuzzy Negations and Automorphisms. *Anais CNMAC* **1984**, *2*, 1127–1129.
22. Weber, S. A General Concept of Fuzzy Connectives, Negations and Implications Based on T-Norms and T-Conorms. *Fuzzy Sets Syst.* **1983**, *11*, 115–134. [CrossRef]
23. Trillas, E. Sobre funciones de negacin en la teora de conjuntos difusos. *Stochastica* **1979**, *III*, 47–60. (In Spanish)
24. Shi, Y. A Deep Study of Fuzzy Implications. PhD Thesis, Ghent University, Ghent, Belgium, 2009.
25. Fodor, J. On fuzzy implication operators. *Fuzzy Sets Syst.* **1991**, *42*, 93–300. [CrossRef]
26. Archer, B.; Weisstein, E.W. Lagrange Interpolating Polynomial. Available online: https://mathworld.wolfram.com/LagrangeInterpolatingPolynomial.html (accessed on 1 September 2023).
27. Bustince, H.; Burillo, P.; Soria, F. Automorphisms, negations and implication operators. *Fuzzy Sets Syst.* **2003**, *134*, 209–229. [CrossRef]
28. Mathews, J.H.; Fink, K.D.; Mathews, J.H. *Numerical Methods Using MATLAB*, 3rd ed.; Prentice Hall: Upper Saddle River, NJ, USA, 1999.
29. Makariadis, S.; Papadopoulos, B. A Fuzzy Implication-Based Approach for Validating Climatic Teleconnections. *Mathematics* **2022**, *10*, 2692. [CrossRef]
30. Makariadis, S.; Souliotis, G.; Papadopoulos, B. Parametric Fuzzy Implications Produced via Fuzzy Negations with a Case Study in Environmental Variables. *Symmetry* **2021**, *13*, 509. [CrossRef]