Seventh International Workshop on IFSs,
Banská Bystrica, Slovakia, 27 Sept. 2011
NIFS 17 (2011), 4, 11–15

Automatic verification of properties of intuitionistic fuzzy connectives via Mathematica

Trifon Trifonov

Faculty of Mathematics and Informatics, Sofia University
e-mail: `triffon@fmi.uni-sofia.bg`

Abstract: Intuitionistic fuzzy logic as defined by K. Atanassov [1, 3], is an extension of fuzzy logic, using the more general intuitionistic fuzzy sets as a model. The extension allows for many different definitions of various logical connectives, such as implication and negation, which can be suitable for different needs. This paper suggests a method for automatic verification of properties of intuitionistic fuzzy connectives using the computer algebra system Mathematica [6].
Keywords: Intuitionistic fuzzy logic, Mathematica, computer algebra, automatic verification
AMS Classification: 03B35, 03B52, 03E72, 68T15, 68W30

1 Introduction

Intuitionistic fuzzy sets (IFS) are an extension of fuzzy sets defined by K. Atanassov [1, 3] that allow for describing uncertainty of propositions. In intuitionistic fuzzy propositional calculus [2] if P is a proposition then its truth value $V(P)$ is represented by the ordered pair

$$V(P) = \langle a, b \rangle, \qquad \text{such that } a, b, a+b \in [0,1].$$

The proposition P is called an *intuitionistic fuzzy tautology* if $a \geq b$. P is called simply a *tautology* if $a = 1$ and $b = 0$.

For easier presentation, below we will use the pairs of variables $\langle a, b \rangle$ and $\langle c, d \rangle$ denoting the truth value of some propositions, instead of the propositions themselves. The classical propositional IFS connectives are defined in [1]:

$$\begin{aligned}
\neg\langle a, b \rangle &:= \langle b, a \rangle, \\
\langle a, b \rangle \wedge \langle c, d \rangle &:= \langle \min(a, c), \max(b, d) \rangle, \\
\langle a, b \rangle \vee \langle c, d \rangle &:= \langle \max(a, c), \min(b, d) \rangle, \\
\langle a, b \rangle \to \langle c, d \rangle &:= \neg\langle a, b \rangle \vee \langle c, d \rangle = \langle \max(b, c), \min(a, d) \rangle
\end{aligned}$$

Notation	Name	$\langle a, b\rangle \to \langle c, d\rangle$
$\to_1$	Zadeh	$\langle \max(b, \min(a, c)), \min(a, d)\rangle$
$\to_2$	Gaines-Rescher	$\langle 1 - \mathrm{sg}(a - c), d.\mathrm{sg}(a - c)\rangle$
$\to_3$	Gödel	$\langle 1 - (1 - c).\mathrm{sg}(a - c), d.\mathrm{sg}(a - c)\rangle$
$\to_4$	Kleene-Dienes	$\langle \max(b, c), \min(a, d)\rangle$
$\to_5$	Łukasiewicz	$\langle \min(1, b + c), \max(0, a + d - 1)\rangle$
$\to_{11}$	Atanassov 1	$\langle 1 - (1 - c).\mathrm{sg}(a - c), d.\mathrm{sg}(a - c).\mathrm{sg}(d - b)\rangle$
$\to_{12}$	Atanassov 2	$\langle \max(b, c), 1 - \max(b, c)\rangle$

Table 1: Examples for IFS implications

2 Examples of variants of intuitionistic fuzzy connectives

In order to present the examples in the following section, we will make use of the "signum" function, defined as follows:

$$\mathrm{sg}(x) = \begin{cases} 1, & \text{if } x > 0, \\ 0, & \text{if } x \leq 0. \end{cases}$$

The most crucial connective determining the meaning of derivability is the logical implication. Already for fuzzy logic there are different means to define its meaning. Table 1 lists some examples for definitions of implications from [4, 7].

Every implication naturally generates a corresponding negation, by defining

$$\neg\langle a, b\rangle = \langle a, b\rangle \to \langle 0, 1\rangle.$$

In [7], two sets of axioms are analysed. The first set consists of the axioms of intuitionistic logic (cf. [9]). The purpose of the verification is to establish the relation between IF logic and intuitionistic logic (cf. [5]). The second set captures the axioms for fuzzy logic and is defined by Klir and Yuan in [8]. The first axiom in this list is

$$(\forall x, y)(x \leq y \to (\forall z)(I(x, z) \geq I(y, z)),$$

where x, y, z are the fuzzy truth values of some arbitrary propositions. In order to verify this axiom for IFS, we need to consider the ordering of truth values as defined in [1]:

$$\langle a, b\rangle \prec \langle c, d\rangle \quad \Longleftrightarrow \quad (a \leq c) \wedge (b \geq d).$$

For both sets of axioms, two questions can be posed: whether the axiom is valid as a tautology or, if not, whether it is valid as an intuitionistic fuzzy tautology.

3 Verifying logical properties with Mathematica

The computer algebra system Mathematica provides a very convenient framework for handling symbolic definitions and includes tools to automatically simplify arithmetic expressions. The formal verification (or refutal) of the axioms from Section 2 in fact amounts to checking the validity of universal properties in real arithmetic in the form $\forall \vec{x} \in \mathbb{R}\ A(\vec{x}) \to B(\vec{x})$, where A and B are conjuncts (prospositional formulas consisting only of conjunctions) that involve the real functions $+, -, ., \min, \max, \mathrm{sg}$ and the predicates $<$ and $=$. All these functions are available in Mathematica, except for sg, which can be defined as

$$\text{Sg[x\_]} = 1 - \text{UnitStep}[-x].$$

The formula A holds all assumptions of the axiom, while the formula B states the conclusion of the axiom.

The most simple tool that can be used to transform algebraic inequalities is the function `Simplify`. It attempts to find the simplest form of an algebraic expression. A more advanced form of this function is `FullSimplify`, which attempts a wider range of simplification techniques.

All properties, which are being checked, operate under the basic IFS conditions that $a, b, a+b \in [0; 1]$. This is captured by the following Mathematica definition:

$$\begin{aligned} \text{ifs[a\_, b\_]} &= 0 \le a \le 1 \ \&\&\ 0 \le b \le 1 \ \&\&\ 0 \le a+b \le 1 \ \&\& \\ & \quad a \in \text{Reals} \ \&\&\ b \in \text{Reals}; \end{aligned}$$

These conditions can be used to further simplify the arithmetic inequality that is being checked. Thus some universal quantifiers can be eliminated based on the conditions in the assumption formula A. This can be done via the Mathematica function `Resolve`.

Sometimes it is quicker to find a counterexample for an axiom rather than trying to verify it. The Mathematica function `FindInstance` can be used on the negation of the axioms, which becomes an existential formula, to compute counterexamples of non-valid axioms.

Some definitions of IFS implication make use of non-continuous functions (such as sg) and functions with a non-continous derivative (such as $\min, \max$). These functions complicate the task of automatic proof of properties. The task can be simplified if we eliminate the use of these functions by considering cases. All their uses can be automatically detected by using the `Cases` function as follows:

$$\text{ListCases[Expr\_]:=Cases[Expr, UnitStep[x\_]|Min[x\_, y\_]|Max[x\_, y\_]}, \{0, \infty\}];$$

Afterwards, we consider separate cases as follows: for every use of `UnitStep[x]`, we consider the cases $\{x \ge 0, x < 0\}$ and for every use of `Min[x,y]` and `Max[x,y]` we consider the cases $\{x - y \ge 0, x - y < 0\}$. We define rewrite rules depending on these cases as follows:

$$\begin{aligned} \text{CalcRules[x\_} \ge 0] &= \{\text{UnitStep}[x] \to 1\}; \\ \text{CalcRules[x\_} < 0] &= \{\text{UnitStep}[x] \to 0\}; \\ \text{CalcRules[x\_} - \text{y\_} \ge 0] &= \{\text{Min}[x, y] \to y, \text{Max}[x, y] \to x\}; \\ \text{CalcRules[x\_} - \text{y\_} < 0] &= \{\text{Min}[x, y] \to x, \text{Max}[x, y] \to y\}; \end{aligned}$$

All possible combinations of cases are generated by using the function `Outer`, which computes an outer tensor product of a second-order tensor containing cases on all possible arguments to $\min, \max$ and sg. Finally, the axiom assumptions A and conclusion B are refined using these rewrite rules using the Mathematica function `Refine`.

4 MathIFS — an implementation of automatic checking of properties of IFS connectives

The implementation of the automatic IFS property checker MathIFS takes advantage of the symbolic and functional features of Mathematica. For example, the implication $\rightarrow_{11}$ from above can be implemented as follows:

$$\text{Atanassov1}[\{\text{a\_}, \text{b\_}\}, \{\text{c\_}, \text{d\_}\}] = \{1 - (1 - c)\text{Sg}[a - c], d\,\text{Sg}[a - c]\text{Sg}[d - b]\};$$

The sample axiom in Section 2 can be defined as

$$\begin{aligned}
\text{Axiom1}[\text{Impl_}] &= \forall_{\{a,b,c,d,e,f\},\text{prec}[\{a,b\},\{c,d\}]\ \&\&\ \text{ifs}[e,f]} \\
&\qquad \text{prec}[\text{Impl}[\{c, d\}, \{e, f\}], \text{Impl}[\{a, b\}, \{e, f\}]]; \\
\text{prec}[\{\text{a_}, \text{b_}\}, \{\text{c_}, \text{d_}\}] &= \text{ifs}[a, b]\ \&\&\ \text{ifs}[c, d]\ \&\&\ a \le c\ \&\&\ b \ge d;
\end{aligned}$$

The actual automatic is defined by alternating between proof (using `Resolve`) and refutal (using `FindInstance`), attempting each for a gradually increasing time interval, until one of them succeeds via the `TimeConstrained` function. This can be automatically performed for a list of axioms and list of implications, using the `Map` function.

In addition, since some of the axioms contain references to a notion of negation or to a notion of tautology, we can explore variants of axioms by considering rewrite rules such as

$$\begin{aligned}
\text{IAxiomModify} &= \{\text{NegF} \rightarrow \text{ImplNeg}[\text{Impl}], \text{Taut} \rightarrow \text{IFTautology}\}; \\
\text{IFTautology}[\{\text{x_}, \text{y_}\}] &= x \ge y; \quad \text{ImplNeg}[\text{Impl_}] = \text{Impl}[\#, \{0, 1\}]\&;
\end{aligned}$$

where ImplNeg defined a negation using the respective implication and IFTautology defines the notion of IF tautology from Section 1.

In addition to checking of axioms, MathIFS can be used to establish relations between different implications and negations, as defined in [7]. For example, consider two implication variants $\rightarrow_a$ and $\rightarrow_b$; we can verify whether

$$\forall a, b, c, d\ \big(\langle a, b\rangle \rightarrow_a \langle c, d\rangle\big)\Box\big(\langle a, b\rangle \rightarrow_b \langle c, d\rangle\big)$$

for some relation $\Box \in \{=, \preceq, \succeq\}$.

5 Conclusion and relation to other work

MathIFS was used to verify most the results which were published in [7]. For some of the implications, especially those containing non-linear terms, or requiring many case splits, the automatic proving process failed to converge, but for most of them, the verification was straightforward.

Another verification approach was followed by D. Dimitrov [10], where the axioms are verified numerically rather than symbolically. Specifically, test sets of random truth values were generated and the validity of the axioms was tested against these sets. An advantage of our approach is that the validity is verified with a higher degree of certainty, but at the cost of more time and memory resources.

References

[1] Atanassov, K. (1983) Intuitionistic fuzzy sets, *VII ITKR's Session, Sofia*, June 1983 (Deposed in Central Sci. - Techn. Library of Bulg. Acad. of Sci., 1697/84) (in Bulg.).

[2] Atanassov, K. (1988) Two variants of intuitonistic fuzzy propositional calculus. *Preprint IM-MFAIS-5-88*, Sofia.

[3] Atanassov, K. (1999) *Intuitionistic Fuzzy Sets: Theory and Applications*. Springer-Physica Verlag, Heidelberg.

[4] Atanassov, K. (2005) Intuitionistic fuzzy implications and Modus Ponens. *Notes on IFS*, Vol. 11, No. 1, 1–4.

[5] Atanassov, K., G. Gargov. (1990) Intuitionistic fuzzy logic. *Comptes Rendus de l'Academie bulgare des Sciences*, Tome 43, 9–12.

[6] Wolfram, S. *Mathematica: A System for Doing Mathematics by Computer*. Addison-Wesley Longman Publishing Co., Inc., 1988, Boston, MA, USA.

[7] Trifonov, T., K. Atanassov. (2006) On some intuitionistic properties of intuitionistic fuzzy implications and negations. In: *Computational Intelligence, Theory and Applications* (Reusch B., Ed.), Vol.38, Advances in Soft Computing, Dortmund, Germany. Springer, Berlin, September 2006, 151–158.

[8] Klir, G., B. Yuan. (1995) *Fuzzy Sets and Fuzzy Logic*, Prentice Hall, New Jersey.

[9] Rasiova, H., R. Sikorski. (1963) *The Mathematics of Metamathematics*, Pol. Acad. of Sci. Warszawa

[10] Dimitrov, D. (2011) IFSTool – software for intuitionistic fuzzy sets. *Issues in IFSs and GNs*, Vol. 9, 61–69.