A New Continuous Propositional Logic

Riccardo Poli, Mark Ryan and Aaron Sloman

School of Computer Science, The University of Birmingham,
Edgbaston, Birmingham B15 2TT, UK

Abstract. In this paper we present Minimal Polynomial Logic (MPL), a generalisation of classical propositional logic which allows truth values in the continuous interval $[0,1]$ and in which propositions are represented by multi-variate polynomials with integer coefficients.
The truth values in MPL are suited to represent the probability of an assertion being true, as in Nilsson's Probabilistic Logic, but can also be interpreted as the degree of truth of that assertion, as in Fuzzy Logic. However, unlike fuzzy logic MPL respects all logical equivalences, and unlike probabilistic logic it does not require explicit manipulation of possible worlds.
In the paper we describe the derivation and the properties of this new form of logic and we apply it to solve and better understand several practical problems in classical logic, such as satisfiability.

1 Introduction

There are many proposals in the literature for extending logic beyond the simple truth values {false, true} or $\{0,1\}$ to truth values in the interval $[0,1]$. Two well-known such extensions are *fuzzy logic* and *probabilistic logic*.

Fuzzy logic [1] is motivated by the wish to express degrees of truth/falsity of propositions. For example, as the property of being tall admits of degrees, fuzzy logic allows the truth value of the sentence 'John is tall' to be some number in the interval $[0,1]$ depending on how tall John is relative to the ambient population. Although fuzzy logic has several important applications, one of its weaknesses is that it does not respect some logical equivalences such as $\neg(x_1 \wedge \neg x_2) \equiv x_2 \vee (\neg x_1 \wedge \neg x_2)$ in the presence of non-binary variables [3].

Nilsson's probabilistic logic [8, 9], on the other hand, is not concerned with inherent degrees of truth, but with the fact that we may have only partial knowledge about the truth or falsity of sentences. In probabilistic logic, a 'truth value' in $[0,1]$ is taken to be the probability that the sentence is true. From the perspective of probabilistic logic, 'John is tall' is either true or it is false, but we may have only partial information about his size and on that basis we may assign to the sentence a number in $[0,1]$ representing the probability that it is true. Despite its clear and precise definition, Nilsson's probabilistic logic requires the explicit computation of the truth or falsity of a proposition in all possible worlds (see section 4 for more details).

In this paper we present a generalisation of classical propositional logic, called *Minimal Polynomial Logic* (MPL), initially developed to facilitate incremental searching for solutions to logical problems, which allows handling continuous truth values in the range $[0, 1]$. The properties of MPL which we will describe suggest that the most suitable interpretation of such truth values is the probability of the assertion being true, as in Nilsson's probabilistic logic. However, unlike probabilistic logic MPL does not require explicit manipulation of all possible worlds.

Despite this probabilistic orientation, for specific applications in which a logic which respects all logical equivalences is required, the truth values of MPL could also represent the degree of truth of the assertion, as in fuzzy logic or fuzzy control. Unfortunately, when this interpretation is adopted, some identities that are universally considered fundamental in fuzzy logic (but not in fuzzy control) do not hold.

The paper is organised as follows. In Section 2 we introduce Polynomial Logics (PLs), which are simple generalisations of classical predicate logic in which propositions are represented by multi-variate polynomials. The simplest form of polynomial logic, which we will denote as PL_0, is the precursor of MPL which is described (Section 3). Some applications of these two types of logic as well as their relations with fuzzy logic and probabilistic logic are discussed in Section 4. We make some final remarks in Section 5.

2 Polynomial Logics

In classic logic the variables x_i which are present in a proposition e can only take two values, 0 and 1. Given the standard definitions of the connectives $\wedge$, $\vee$ and $\neg$ (e.g. $x_1 \wedge x_2 = 1$ iff $x_1 = x_2 = 1$), the same is true of the values taken by e. One way of generalising this kind of binary (or Boolean) logic would be to consider expressions with variables that can take continuous values between 0 (false) and 1 (true) and to generalise the ordinary logic connectives.

A natural way of generalising such connectives is to consider functions that can fit the datapoints represented by the truth tables of the ordinary connectives. For example, if we want to generalise the $\vee$ function, $x_1 \vee x_2$, we have to select a function $o(x_1, x_2)$ such that $o(0, 0) = 0$, $o(0, 1) = 1$, $o(1, 0) = 1$ and $o(1, 1) = 1$.

A simple form of such functions is obtained by using polynomials that can fit the truth tables of the ordinary logic connectives. For example, the polynomials $a(x_1, x_2) = \frac{1}{4}x_1x_2(1 + x_1)(1 + x_2)$, $o(x_1, x_2) = 1 - a(1 - x_1, 1 - x_2) = 1 - \frac{1}{4}(1 - x_1)(1 - x_2)(2 - x_1)(2 - x_2)$ and $n(x_1) = (1 - x_1)(1 + x_1)$ generalise the logical connectives $\wedge$, $\vee$ and $\neg$, respectively.[1] There are infinitely many such generalisations.

Having defined a set of generalised connectives any ordinary logic expression e can be generalised by simply replacing the ordinary connectives with the generalised ones. With polynomial connectives an entire class of polynomial propositional logics, PL, can thus be defined.

---

[1] Other connectives such as $\rightarrow$ and $\leftrightarrow$ can be obtained likewise using standard equivalences.

The *lowest degree* polynomials that can fit the truth-table entries of the ordinary logic connectives,

$$\begin{aligned} o(x_1, x_2) &= x_1 \vee x_2 = 1 - (1 - x_1)(1 - x_2), \\ a(x_1, x_2) &= x_1 \wedge x_2 = x_1 x_2, \\ n(x_1) &= \neg x_1 = 1 - x_1, \end{aligned} \tag{1}$$

define the most parsimonious (lowest-degree) polynomial logic which we will denote with the symbol PL_0. More formally:

Definition 1. Given a propositional formula e, its PL_0 version e_p is the polynomial obtained by replacing the ordinary connectives with those given in Eq. 1.

Example 1. Consider the expression $e = (x_1 \vee (x_2 \wedge \neg x_3)) \wedge (x_1 \rightarrow x_2)$. The PL_0 version of it is:

$$\begin{aligned} e_p &= (1 - (1 - x_1)(1 - (x_2(1 - x_3))))(1 - x_1(1 - x_2)) \\ &= 2x_1x_2x_3 - {x_1}^2x_2x_3 + {x_1}^2{x_2}^2x_3 - 2x_1x_2 + 2{x_1}^2x_2 - \\ &\quad -{x_1}^2{x_2}^2 - x_2x_3 - {x_2}^2x_3x_1 + x_1 - {x_1}^2 + x_2 + x_1{x_2}^2. \end{aligned}$$

PL_0 and classical logic give the same truth values when the propositional variables take the values 0 and 1.

Theorem 2. $\forall x_i \in \{0, 1\}$, $e = e_p$.

Proof. Since the polynomials $o(x_1, x_2)$, $a(x_1, x_2)$ and $n(x)$, when evaluated with $x_i \in \{0, 1\}$, take the same values of their discrete (binary/Boolean) counterparts, this is also true for the expression e_p. □

Example 2. If the original expression e is in conjunctive normal form (CNF), i.e. a conjunction ($\wedge$) of disjunctions ($\vee$) of literals (variables or negated variables) of the form

$$e = \bigwedge_{i=1}^{M} \left(\bigvee_{j=1}^{K_i} l_{ij} \right), \tag{2}$$

where $l_{ij} \in \{x_1, \cdots, x_N, \neg x_1, \cdots, \neg x_N\}$, then its PL_0 version is given by:

$$e_p = \prod_{i=1}^{M} \left(1 - \prod_{j=1}^{K_i} (1 - l_{c,ij}) \right), \tag{3}$$

where $l_{c,ij} \in \{x_1, \cdots, x_N, (1 - x_1), \cdots, (1 - x_N)\}$. The fact that $e_p = 1$ iff $\forall i \, \exists j : l_{c,ij} = 1$ clarifies the equivalence between e and e_p in the case of binary variables.

3 Minimal Polynomial Logic

In the previous section we have introduced the notion of polynomial logics in general and described PL_0 in particular. In this section we will obtain from PL_0 a new form of continuous logic that we call Minimal Polynomial Logic (MPL).

Definition 3. Given a propositional formula e, its MPL version e_m is obtained from the PL_0 version e_p by distributing $+$ over $\times$ throughout and then substituting subexpressions of the form x_i^k (with $k > 1$) with x_i. This substitution will be sometimes be denoted with $(\cdot)_m$.

Example 3. Let us consider the exclusive or function: $e = (x_1 \wedge \neg x_2) \vee (\neg x_1 \wedge x_2)$. Its PL_0 and MPL versions are $e_p = x_1 + x_2 - 3x_1x_2 + x_1^2x_2 + x_1x_2^2 - x_1^2x_2^2$ and $e_m = x_1 + x_2 - 2x_1x_2$, respectively.

This simple substitution is one of the main ideas in this paper. As will be seen, it has significant consequences (e.g. Thm. 5).

As before this logic agrees with classical logic on the Boolean truth values:

Theorem 4. $\forall x_i \in \{0, 1\}$, $e_p = e_m = e$.

Proof. If $x_i \in \{0, 1\}$ then $x_i^k = x_i$ $(k > 1)$, therefore the substitution given in Def. 3 does not change the value of e_p. $e_m = e$ follows from Thm. 2. □

However, MPL has an important property which distinguishes it from other PLs:

Theorem 5. *Two propositions e and e' are logically equivalent iff their MPL versions e_m and e'_m are the same polynomial.*

Proof. $\Leftarrow$ If $e_m \equiv e'_m$ then, in particular, $\forall x_i \in \{0, 1\}$ $e_m = e'_m$. Thus, by Thm. 4 $e \equiv e'$.
$\Rightarrow$ Suppose $e_m \not\equiv e'_m$, then there exist some coefficients $c_i \neq 0$ such that

$$e_m - e'_m = c_1 x_{k_1^1} \cdots x_{k_{L_1}^1} + \cdots + c_D x_{k_1^D} \cdots x_{k_{L_D}^D}.$$

Let c_m be the coefficient of any term of minimal degree. Set the variables which occur in that term to 1 and all the other variables occurring either in e_m or e'_m to 0. Then $e_m - e'_m = c_m \neq 0$, so by Thm. 4 $e \neq e'$ for that assignment. □

Corollary 6. *1. e is satisfiable iff $e_m \not\equiv 0$. Moreover, the second part of the proof of Thm 5 gives an assignment making e true.*
2. e is a tautology iff $e_m \equiv 1$. Moreover, if $e_m \not\equiv 1$ then the second part of the proof of Thm 5 gives an assignment making e false.

Example 4. Let us consider again the expression $e = (x_1 \vee (x_2 \wedge \neg x_3)) \wedge (x_1 \rightarrow x_2)$. The MPL version of it is:

$$e_m = x_1x_2x_3 - x_2x_3 + x_2.$$

The lowest degree term of e_m is x_2, therefore, according to the procedure outlined in the proof of Theorem 5, the assignment $x_1 = 0$, $x_2 = 1$, $x_3 = 0$ satisfies e. This is correct, as

$$e = (0 \vee (1 \wedge \neg 0)) \wedge (0 \rightarrow 1) = (0 \vee (1 \wedge 1)) \wedge 1 = (0 \vee 1) \wedge 1 = 1 \wedge 1 = 1.$$

Example 5. Let us now consider the expression $e = x_1 \wedge x_2 \wedge (\neg x_1 \vee \neg x_2)$. The PL_0 version of it is:

$$e_p = x_1 x_2 (1 - x_1 x_2) = x_1 x_2 - x_1^2 x_2^2,$$

while its MPL version is

$$e_m = x_1 x_2 - x_1 x_2 \equiv 0$$

which shows that e is unsatisfiable. This is correct as can be readily seen by rewriting $e = e' \wedge \neg e'$ with $e' = x_1 \wedge x_2$.

This result gives a new and interesting way of checking entailment between propositional formulas:

Corollary 7. $e \models e'$ *iff* $e_m \equiv (e_m e'_m)_m$.

Proof. $e \models e'$ iff $e \rightarrow e' \equiv \top$ iff $(1 - e_m(1 - e'_m))_m \equiv 1$ iff $e_m \equiv (e_m e'_m)_m$. □

Example 6. Let us consider the expressions $e = (x_1 \vee x_2) \wedge (\neg x_2 \vee x_3)$ and $e' = x_1 \vee x_3$. We want to check if e entails e'. As $e_m = x_2 x_3 + x_1 - x_1 x_2$ and $e'_m = x_1 + x_3 - x_1 x_3$, simple calculations can show that $e_m \equiv (e_m e'_m)_m$.

The next two lemmas are used for the following decomposition theorem 10 and Thm. 17.

Lemma 8. *Let P_1, P_2 be polynomials.*

1. $(P_1 + P_2)_m \equiv (P_1)_m + (P_2)_m$.
2. $(P_1 P_2)_m \equiv (P_1)_m (P_2)_m$ *if P_1 and P_2 have no variables in common.*

Proof. 1. Suppose x_i^k is a subexpression in $P_1 + P_2$, then it is a subexpression in P_1 or P_2 or both, and so will be reduced to x_i in $(P_1)_m + (P_2)_m$. 2. Suppose P_1 and P_2 have no variables in common and x_i^k is a subexpression in $P_1 P_2$, then it is a subexpression in P_1 or P_2, and so will be reduced to x_i in $(P_1)_m (P_2)_m$. □

Lemma 9. $e_m \equiv x_1 (e[\top/x_1])_m + (1 - x_1)(e[\bot/x_1])_m$.

Proof. First note that $e \equiv (x_1 \wedge e[\top/x_1]) \vee (\neg x_1 \wedge e[\bot/x_1])$ therefore:

$$\begin{aligned}
e_m &\equiv ((x_1 \wedge e[\top/x_1]) \vee (\neg x_1 \wedge e[\bot/x_1]))_m && \text{Thm. 5}\\
&\equiv (1-(1-x_1 e[\top/x_1]_p)(1-(1-x_1)e[\bot/x_1]_p))_m \\
&\equiv (x_1 e[\top/x_1]_p + (1-x_1)e[\bot/x_1]_p \\
&\quad + x_1(1-x_1)e[\top/x_1]_p e[\bot/x_1]_p)_m \\
&\equiv (x_1 e[\top/x_1]_p)_m + ((1-x_1)e[\bot/x_1]_p)_m \\
&\quad + (x_1(1-x_1)e[\top/x_1]_p e[\bot/x_1]_p)_m && \text{Lemma 8}\\
&\equiv x_1(e[\top/x_1]_p)_m + (1-x_1)(e[\bot/x_1]_p)_m && \text{Lemma 8}\\
& && x_1 \text{does not occur in } e[\cdot/x_1]\\
& && (x_1(1-x_1))_m \equiv 0\\
&\equiv x_1 e[\top/x_1]_m + (1-x_1)e[\bot/x_1]_m
\end{aligned}$$

□

The following theorem shows how an MPL expression can be decomposed as a linear combination of orthogonal basis of MPL expressions.

Theorem 10. *$e_m = \sum_{i=1}^{2^N} y_i (e_i)_m$, where*

$$\begin{aligned}
y_1 &= x_1 x_2 \cdots x_N,\\
y_2 &= (1-x_1)x_2 \cdots x_N,\\
y_3 &= x_1(1-x_2)\cdots x_N,\\
&\cdots\\
y_{2^N} &= (1-x_1)(1-x_2)\cdots(1-x_N),
\end{aligned}$$

are an orthogonal basis for MPL with the scalar product $\langle y_i, y_j\rangle = (y_i y_j)_m$ and

$$\begin{aligned}
e_1 &= e[\top/x_1, \top/x_2, \ldots, \top/x_N],\\
e_2 &= e[\bot/x_1, \top/x_2, \ldots, \top/x_N],\\
e_3 &= e[\top/x_1, \bot/x_2, \ldots, \top/x_N],\\
&\cdots\\
e_{2^N} &= e[\bot/x_1, \bot/x_2, \ldots, \bot/x_N].
\end{aligned}$$

Proof. Apply Lemma 9 recursively to all the variables in e. □

Using the results just introduced, we are now able to give an alternative characterisation of entailment:

Theorem 11. *$e \models e'$ iff $e_m \le e'_m$, $\forall x_i \in [0,1]$.*

Proof. $\Leftarrow$ immediate.
$\Rightarrow$ $e_m - e'_m = \sum_i y_i((e_i)_m - (e'_i)_m) \le 0$ as $(e_i)_m \le (e'_i)_m$. □

4 Applications and Relations with Other Logics

4.1 Use and Interpretations of PL_0

In addition of being the precursor of MPL, PL_0 can have practical applications on its own.

Algebraic Logical Calculus. As a first application, PL_0 can be used to study or to teach classical logic by using only (or mostly) familiar algebraic techniques. The two theorems and the corollary given in this section are an example of this.[2]

The following definition and lemma are required for the next two theorems.

Definition 12. The dual $\hat{e}$ of e is the expression obtained by exchanging $\wedge$ with $\vee$ and $\bot$ with $\top$ in e.

Lemma 13. *e is unsatisfiable iff its dual $\hat{e}$ is a tautology*

Proof. The duality theorem [11, p.26] states that any two expressions e and e' are logically equivalent iff their duals $\hat{e}$ and $\hat{e'}$ are logically equivalent. Therefore, e is unsatisfiable iff $e \equiv \bot$ iff $\hat{e} \equiv \top$. □

Theorem 14. *Let e be a proposition in CNF such as Equation 2. e is unsatisfiable iff $\forall (x_1, \ldots, x_N) \in \{0,1\}^N, \exists i \forall j\, l_{ij} = 1$.*

Proof. If $\hat{e}$ is the dual of e, i.e. $\hat{e} = \bigvee_{i=1}^{M} \left(\bigwedge_{j=1}^{K_i} l_{ij} \right)$, then

$$\hat{e}_p = 1 - \prod_{i=1}^{M} \left(1 - \prod_{j=1}^{K_i} l_{c,ij} \right) .$$

By Lemma and Thm. 2, e is unsatisfiable iff $\forall (x_1, \ldots, x_N) \in \{0,1\}^N\, \hat{e}_p = 1$ iff $\forall (x_1, \ldots, x_N) \in \{0,1\}^N, \exists i \prod_{j=1}^{K_i} l_{c,ij} = 1$ iff $\forall (x_1, \ldots, x_N) \in \{0,1\}^N, \exists i \forall j\, l_{ij} = 1$. □

Corollary 15. *Let e be a proposition in CNF.*

1. *If $\forall i \exists j$ such that $l_{ij} \in \{\neg x_1, \cdots, \neg x_n\}$ then e is satisfiable.*
2. *If $\forall i \exists j$ such that $l_{ij} \in \{x_1, \cdots, x_n\}$ then e is satisfiable.*

Proof. For 1. $(x_1, \ldots, x_N) = (0, \ldots, 0)$ and for 2. $(x_1, \ldots, x_N) = (1, \ldots, 1)$. □

Theorem 16. *Let e be a proposition in Disjunctive Normal Form (DNF), i.e. $e = \bigvee_{i=1}^{M} \left(\bigwedge_{j=1}^{K_i} l_{ij} \right)$. e is unsatisfiable iff $\forall (x_1, \ldots, x_N) \in \{0,1\}^N, \forall i \exists j\, l_{ij} = 1$.*

Proof. The PL_0 version of the dual $\hat{e}$ of e is $\hat{e}_p = \prod_{i=1}^{M} \left(1 - \prod_{j=1}^{K_i} (1 - l_{c,ij}) \right)$. $\forall (x_1, \ldots, x_N) \in \{0,1\}^N\, \hat{e}_p = 1$ iff $\forall (x_1, \ldots, x_N) \in \{0,1\}^N, \forall i \exists j\, l_{ij} = 1$. □

[2] Of course, there are direct proofs based on classical logic only for the results obtained with PL_0.

Relations with Probability. If we interpret the variables occurring in the polynomial e_p as probabilities of being true of the corresponding atomic propositions in e, then the value taken by e_p can be interpreted as the probability that e is true.

To illustrate this, let us consider the expression $e = x_1 \vee x_2$ and imagine that x_1, x_2 and consequently e are stochastic binary variables. If we denote with $\mathcal{P}(x_1)$, $\mathcal{P}(x_2)$ and $\mathcal{P}(e)$ the probability of the events $\{x_1 = 1\}$, $\{x_2 = 1\}$ and $\{e = 1\}$, then on the hypothesis that x_1 and x_2 are independent variables we can write:

$$
\begin{aligned}
\mathcal{P}(e) &= \Pr\{e = 1\} \\
&= \Pr\{x_1 \vee x_2 = 1\} \\
&= \Pr\{x_1 = 1\} + \Pr\{x_2 = 1\} - \Pr\{x_1 = 1\}\Pr\{x_2 = 1\} \\
&= \mathcal{P}(x_1) + \mathcal{P}(x_2) - \mathcal{P}(x_1)\mathcal{P}(x_2) \\
&= 1 - (1 - \mathcal{P}(x_1))(1 - \mathcal{P}(x_2))
\end{aligned}
$$

This expression is formally identical to the PL_0 form of e, namely $e_p = 1 - (1 - x_1)(1 - x_2)$, provided that e_p, x_1 and x_2 are interpreted as the probability of being true of the related binary counterparts. The same observation is valid for the $\neg$ and $\wedge$ polynomial functions.

However, as already mentioned in this example, the probabilistic interpretation of the polynomial connectives is correct only on the hypothesis of independent arguments. As a result, the probabilistic interpretation of e_p is correct if no variable occurs more than once in e. Nonetheless, in many cases e_p can be considered as a reasonable approximation of the exact probability and therefore used for many practical purposes. An example of this is given is in the following subsection.

Towards an explanation for GSAT. The problem of deciding if a proposition is satisfiable is a well known NP-complete problem for which time required for exact solutions is an exponential function of the number of variables [2]. This imposes a serious limit to the number of variables of the expression to be checked. For example, it is reported in the literature that one of the best known exact algorithms for satisfiability checking, the Davis-Putnam procedure [2], cannot practically handle expressions with more than a few hundred of variables [10].

Recently a new, very promising approach to the solution of hard satisfiability problems has been proposed which is based on greedy local search procedures (GSAT) [10, 5]. Given an expression e in CNF such as Eq. 2, GSAT works as follows:

1. Randomly initialise the variables in e.
2. If $e = \top$ then return($\top$).
3. Select a variable such that a change in its truth assignment gives the largest increase in the total number of clauses of e that are satisfied and reverse its assignment.
4. Iterate steps 2–3 for N_{flips} times.
5. Iterate steps 1–4 for N_{tries} times.

This procedure allows finding solutions for satisfiability problems including several hundred (or even thousands) of variables. Although a theoretical analysis of the the algorithm has been undertaken [5], the reason why the simple optimisation of the number of true clauses in an expression leads so frequently to finding an assignment that satisfies such an expression is actually not completely understood. PL_0 provides a possible explanation for this.

If e_p is the PL_0 version of an expression e in CNF such as Eq. 2, then

$$\log(e_p) = \sum_{i=1}^{M} \log\left(\bigvee_{j=1}^{K_i} l_{ij}\right)_p$$

Note that $\log\left(\bigvee_{j=1}^{K_i} l_{ij}\right)_p \in [0, -\infty]$. However, to understand GSAT we imagine that $\log(0) = -K$, for some suitably large number K. On this hypothesis, given any (binary) assignment of the variables,

$$\log(e_p) = -K \times M_\perp = K \times (M_\top - M),$$

$M_\top$ and $M_\perp$ being the number of true and false clauses in e, respectively. Being the logarithm a monotonic increasing function, the probabilistic interpretation of this equation is: *maximising the number of true clauses in e (e.g. using the GSAT algorithm) is equivalent to maximising an approximation (e_p) of the probability of being true of e in the corners of the hypercube* $[0, 1]^N$. Searching for the maxima of e_p moving only on the corners of the hypercube is overconstraining, and GSAT can therefore be generalised and improved by using any optimisation procedure (e.g. gradient ascent or a genetic algorithm) working in $[0, 1]^N$.

Relations with Fuzzy Logic. If we interpret the variables occurring in the polynomial e_p as the degree of truth of the corresponding atomic propositions in e, then the value taken by e_p can be interpreted as the degree of truth of e. In this sense, PL_0 is actually equivalent to a well-known form of fuzzy logic which is often used in fuzzy control [6]. The disadvantages of PL_0 are: a) unlike min/max-based fuzzy logic, it does not respect idempotency properties ($x_1 \wedge x_1 \equiv x_1$ and $x_1 \vee x_1 \equiv x1$), b) like fuzzy logic, it fails to respect some other logical equivalences such as

$$\begin{aligned}(\neg(x_1 \wedge \neg x_2))_p &\equiv 1 - x_1(1 - x_2) \\ &\not\equiv 1 - x_1 - x_2 + 2x_1x_2 + x_2^2 - x_1x_2^2 \\ &\equiv (x_2 \vee (\neg x_1 \wedge \neg x_2))_p.\end{aligned}$$

An advantage of PL_0 as a fuzzy logic is that it is minimally sensitive to errors in the estimation of the degrees of truth of atomic sentences [7].

4.2 Use and Interpretations of MPL

The examples given in Section 3 show how MPL can be used to effectively and naturally answer questions about satisfiability and entailment in classical logic by using algebraic manipulations.

As in the case of MPL, the variables in e_m can be interpreted either as probabilities or fuzzy truth values. In the following we will show how in the first case MPL overcomes all the independency requirements of PL_0, while in the second case it further departs from the usual features of min/max fuzzy logic.

Relations with Probability. The probabilistic interpretation of MPL requires additional work carried out in the following theorem.

Theorem 17. $\mathcal{P}(e) = e_m[\mathcal{P}(x_i)/x_i]$.

Proof. Induction on the number of variables in e.
Base case: 0 variables. Trivial.
Inductive case: Suppose there are k variables in e and the theorem holds for all expressions with $k-1$ variables. Let x_1 be any variable.

$$\begin{aligned}
\mathcal{P}(e) &= \Pr\{x_1 = 1\}\Pr\{e = 1 \mid x_1 = 1\} \\
&\quad + \Pr\{x_1 = 0\}\Pr\{e = 1 \mid x_1 = 0\} \\
&= \mathcal{P}(x_1)\mathcal{P}(e[\top/x_1]) + (1 - \mathcal{P}(x_1))\mathcal{P}(e[\bot/x_1]) \\
&= \mathcal{P}(x_1)(e[\top/x_1])_m[\mathcal{P}(x_i)/x_i] \\
&\quad +(1 - \mathcal{P}(x_1))(e[\bot/x_1])_m[\mathcal{P}(x_i)/x_i] && \text{Ind. Hyp.} \\
&= (x_1 e[\top/x_1]_m + (1 - x_1)e[\bot/x_1]_m)[\mathcal{P}(x_i)/x_i] \\
&= e_m[\mathcal{P}(x_i)/x_i] && \text{Lemma 9}
\end{aligned}$$

□

Example 7. If $e = (x_1 \vee (x_2 \wedge \neg x_3)) \wedge (x_1 \rightarrow x_2)$, then the probability of e being true is $\mathcal{P}(e) = e_m[\mathcal{P}(x_i)/x_i] = (x_1x_2x_3 - x_2x_3 + x_2)[\mathcal{P}(x_i)/x_i] = \mathcal{P}(x_1)\mathcal{P}(x_2)\mathcal{P}(x_3) - \mathcal{P}(x_2)\mathcal{P}(x_3) + \mathcal{P}(x_2)$.

As clarified by the previous results, MPL yields the correct probability of an expression being true, even in the case of dependent subexpressions (i.e. reused variables).

Relations with Nilsson's Probabilistic Logic. In probabilistic logic, each world w_i is an assignment for the variables present in a proposition e to which a probability p_i of being the case is associated. The probability of e being true is then represented by

$$\Pr\{e = 1\} = \sum_i p_i w_i(e), \tag{4}$$

where $w_i(e)$ is the result of evaluating e in w_i. This expression shows that Nilsson's probabilistic logic requires the explicit computation of the truth or falsity of a proposition in all possible worlds.

The relation between probabilistic logic and (the probabilistic interpretation of) MPL is clarified by the following

Corollary 18. $\mathcal{P}(e) = \sum_{i=1}^{2^N} y_i[\mathcal{P}(x_i)/x_i](e_i)_m$ *where* y_i *and* e_i *are defined as in Thm. 10.*

Proof. Apply Thm. 17 to e_m expressed as in Thm. 10. □

By considering for example that $y_1[\mathcal{P}(x_i)/x_i] = \mathcal{P}(x_1 \wedge x_2 \wedge \ldots \wedge x_N) = \Pr\{x_1 = \top, x_2 = \top, \ldots, x_N = \top\}$, it can be easily understood that $y_i[\mathcal{P}(x_i)/x_i] = \Pr\{w_i\} = p_i$. On the other hand $(e_i)_m = w_i(e)$, and therefore the last corollary can be reformulated as

$$\mathcal{P}(e) = \sum_{i=1}^{2^N} p_i w_i(e),$$

which is exactly the same expressions as in Eq. 4.

This clarifies how (the probabilistic interpretation of) MPL generalises probabilistic logic as the atoms it adopts are are not entire worlds but the sentences composing such worlds.

Relations with Fuzzy Logic. Let us now reconsider the interpretation of MPL as fuzzy logic. Thm. 5 guarantees that MPL respects logical equivalence. For example, $(x_1 \wedge x_1)_m \equiv x_1 \equiv (x_1)_m$, $(x_1 \vee x_1)_m \equiv x_1 \equiv (x_1)_m$,

$$\begin{aligned}(\neg(x_1 \wedge \neg x_2))_m &\equiv 1 - x_1 + x_1 x_2 \\ &\equiv (x_2 \vee (\neg x_1 \wedge \neg x_2))_m,\end{aligned}$$

$(x_1 \wedge \neg x_1)_m \equiv 0 \equiv (\bot)_m$ and $(x_1 \vee \neg x_1)_m \equiv 1 \equiv (\top)_m$. Note that the last three equivalences are not valid in the various forms of fuzzy logic.

However, while on the one hand the fuzzy interpretation of MPL seems to have better properties than fuzzy logic, on the other hand it departs even more than PL_0 from the behaviour of the standard min/max fuzzy logic. An example of this is the expression $x_1 \wedge \neg x_1$ which evaluates to something in $[0.5, 1]$ in fuzzy logic, to something in $[0, 0.5]$ in PL_0, and to 0 in MPL. This would certainly be considered an anomalous result if the expression represents the degree of truth of the fact that some property is partly present and partly not present at the same time.

5 Conclusions

In this paper we have presented minimal polynomial logic, a generalisation of classical propositional logic which allows continuous truth values.

In its non-minimal form PL_0, our logic can be used either as a fuzzy logic or as an approximate probabilistic logic. We have used this form of logic to prove some results about classical logic, which are transparent in MPL. The proofs of such results are based on a natural integration of calculus and standard logical techniques. In addition, with a simple logarithm transformation PL_0 provides a long-sought explanation for the enigmatic GSAT algorithm [4].

MPL has all these properties but it also respects logical equivalence (Theorem 5). This means that whatever we can prove to be true for MPL, for example using calculus, is true in classical logic and vice versa. An application of this theorem, Corollary 6, provides a new way of checking the satisfiability of a proposition based only on algebraic manipulations. Thanks to Cor. 7 and Thm. 11, the same is also true for checking entailment.

Finally, the probabilistic interpretation of MPL, supported by Thm. 17, gives the probability of a proposition being true even in the case in which there are repeated variables. This does not require the explicit evaluation of the expression in all possible worlds needed by Nilsson's probabilistic logic. However, Thm.10 guarantees that the probabilities computed with MPL and probabilistic logic are the same.

Acknowledgements

The authors thank Alan P. Sexton of the School of Computer Science, The University of Birmingham, for useful discussions and suggestions. The second author acknowledges partial support from Esprit WG ModelAge (8319).

References

1. J. C. Bezdek. Fuzzy models – what are they and why? *IEEE Transactions on Fuzzy Systems*, 1(1):1–6, 1993.
2. M. D. Davis and E. J. Weyuker. *Computability, Complexity and Languages*. Academic Press, London, 1983.
3. C. Elkan. The paradoxical success of Fuzzy Logic. *IEEE Expert*, 9(4):3–8, August 1994.
4. I. Gent and T. Walsh. The enigma of SAT hill-climbing procedures. Technical Report 605, Department of Artificial Intelligence, University of Edinbrugh, 1992.
5. I. P. Gent and T. Walsh. An empirical analysis of search in GSAT. *Journal of Artificial Intelligence Research*, 1:47–59, 1993.
6. V. Kreinovich, C. Quintana, R. Lea, O. Fuentes, A. Lokshin, S. Kumar, L. Boricheva, and L. Reznik. What non-linearity to choose? A mathematical foundation of fuzzy control. In *Proceedings International Conference on Fuzzy Systems and Intelligent Control*, pages 349–412, Louisville, KY, 1992.
7. H. T. Nguyen, V. Kreinovich, and D. Tolbert. A measure of average sensitivity for fuzzy logics. *International Journal of Uncertainty, Fuzziness, and Knowledge-Based Systems*, 2(4):361–375, 1994.
8. N. J. Nilsson. Probabilistic logic. *Artificial Intelligence*, 28:71–87, 1986.
9. N. J. Nilsson. Probabilistic logic revisited. *Artificial Intelligence*, 59(1–2):39–42, 1993.
10. B. Selman, H. Levesque, and D. Mitchel. A new method for solving hard satisfiability problems. In *Proc. 10-th National Conference of Artificial Intelligence AAAI'92*, pages 440–446, San Jose, CA, July 1992.
11. D. van Dalen. *Logic and Structure*. Universitext. Springer Verlag, third edition, 1994.

This article was processed using the LaTeX macro package with LLNCS style