národní
úložiště
šedé
literatury

Modeling Costs of Program Runs in Fuzzified Propositional Dynamic Logic

Běhounek, Libor
2008

Dostupný z http://www.nusl.cz/ntk/nusl-39082

Dílo je chráněno podle autorského zákona č. 121/2000 Sb.

Tento dokument byl stažen z Národního úložiště šedé literatury (NUŠL).

Datum stažení: 23.04.2024

Další dokumenty můžete najít prostřednictvím vyhledávacího rozhraní nusl.cz .

Modeling Costs of Program Runs in Fuzzified Propositional Dynamic Logic

Post-Graduate Student:
MGR. LIBOR BĚHOUNEK

Institute of Computer Science of the ASCR, v. v. i.
Pod Vodárenskou věží 2

182 07 Prague, Czech Republic

behounek@cs.cas.cz

Supervisor:
DOC. PHDR. PETR JIRKŮ, CSC.

Faculty of Arts
Charles University in Prague
Celetná 20

116 42 Prague, Czech Republic

petr.jirku@ff.cuni.cz

Field of Study:
Logic

The work was supported by grant No. IAA900090703 *Dynamic Formal Systems* of the Grant Agency of the Academy of Sciences of the Czech Republic and Institutional Research Plan No. AV0Z10300504. The advisor for my research in the area of fuzzy logic is Prof. RNDr. Petr Hájek, DrSc. I have profitted from discussions with Marta Bílková and Petr Cintula.

Abstract

The paper introduces a logical framework for representing costs of program runs in fuzzified propositional dynamic logic. The costs are represented as truth values governed by the rules of a suitable t-norm fuzzy logic. A translation between program constructions in dynamic logic and fuzzy set-theoretical operations is given, and the adequacy of the logical model to the informal motivation is demonstrated. The role of tests of conditions in programs is discussed from the point of view of their costs, which hints at the necessity of distinguishing between the fuzzy modalities of admissibility and feasibility of program runs.

1. Introduction

It has been argued in [1] that t-norm fuzzy logics can be interpreted as logics of resources or costs, besides their usual interpretation as logics of partial truth. Particular instances of costs are the costs of program runs: typically, a run of a program needs various kinds of resources like machine time for performing instructions, operational memory or disk space for data, access to peripheries or special computation units, etc. Depending on the amount of the resources needed, some runs of programs can be more costly than others. The most usual logical model of programs and program runs is presented by propositional dynamic logic, which will be used as a basis for the present generalization. The aim of this paper is to sketch a logical framework for handling the costs of program runs by means of fuzzy logic, with programs modeled abstractly in propositional dynamic logic, and present some basic observations on the proposed model.

The paper has the following structure: A brief description of t-norm fuzzy logics and their cost-based interpretation is given in Sections 2 and 3. The apparatus of propositional dynamic logic is recalled in Section 4. A combination of these approaches, leading to a model of costs of program runs in fuzzified propositional dynamic logic, is given in Section 5. The role of tests of conditions in programs, which necessitates distinguishing the feasibility and admissibility of program runs in fuzzified propositional dynamic logic, is discussed in Section 6.

It should be noted that the paper only presents an initial sketch of the proposed approach to logical modeling of program costs. The work on this approach is currently in progress and a more comprehensive elaboration is being prepared, with Marta Bílková and Petr Cintula as co-authors.

2. T-norm fuzzy logic

In this section we give a short overview of the most important t-norm fuzzy logics that will be needed later on. Only the standard semantics of t-norm fuzzy logics is presented here, as it suffices for the needs of this paper. For more details on t-norm logics, including their axiomatic systems and general semantics, see [2, 3].

In the standard semantics, formulae of t-norm fuzzy logics are evaluated truth-functionally in the real unit interval $[0, 1]$; i.e., propositional connectives

Institucionální repozitář AV ČR http://hdl.handle.net/11104/0163282

are semantically realized by operations on $[0, 1]$. In particular, the connective called *strong conjunction* & is in t-norm fuzzy logics realized by a *left-continuous t-norm*, i.e., a left-continuous binary operation on $[0, 1]$ which is commutative, associative, monotone, and has 1 as its neutral element. The most important (left-) continuous t-norms are

$x *_{\mathrm{G}} y = \min(x, y)$	Gödel t-norm
$x *_{\Pi} y = x \cdot y$	product t-norm
$x *_{\text{Ł}} y = \max(0, x + y - 1)$	Łukasiewicz t-norm

Every left-continuous t-norm $*$ has a unique *residuum* $\Rightarrow_*$, defined as

$$x \Rightarrow_* y = \sup\{z \mid z * x \leq y\},$$

which interprets *implication* $\rightarrow$ in the logic $\mathrm{L}(*)$ of the left-continuous t-norm $*$. If $x \leq y$, then $x \Rightarrow_* y = 1$; for $x > y$ the residua of the above three t-norms evaluate as follows:

$$x \Rightarrow_{\mathrm{G}} y = y$$
$$x \Rightarrow_{\Pi} y = y/x$$
$$x \Rightarrow_{\text{Ł}} y = \min(1, 1 - x + y)$$

Further propositional connectives of $\mathrm{L}(*)$ are interpreted in the following way:

- *Negation* $\neg$ as $\neg_* x = x \Rightarrow_* 0$
- *Equivalence* $\leftrightarrow$ as
$$x \Leftrightarrow_* y = \min(x \Rightarrow_* y, y \Rightarrow_* x)$$
- *Disjunction* $\vee$ as the maximum, and
- *Weak conjunction* $\wedge$ as the minimum

Optionally, the *delta connective* Δ is added to $\mathrm{L}(*)$ with standard interpretation $\Delta x = 1$ if $x = 1$, and $\Delta x = 0$ otherwise. (We shall always use t-norm logics with Δ in this paper.) The algebra

$$[0, 1]_* = \langle [0, 1], *, \Rightarrow, \vee, \wedge, 0, \Delta \rangle$$

defining an interpretation of propositional t-norm logic is called the *t-algebra* of $*$ (with Δ).

Formulae that always evaluate to 1 are called *tautologies* of the logic $\mathrm{L}(*)$. The formulae that are tautologies of $\mathrm{L}(*)$ for all $*$ from some class $\mathcal{K}$ of left-continuous t-norms form the t-norm logic of the class $\mathcal{K}$. In particular, Hájek's [2] logic BL is the logic of all *continuous* t-norms and the logic MTL of [3] is the logic of all *left*-continuous t-norms: these general logics capture rules valid independently of a particular t-norm realization of &. The proofs in this paper will be carried out in the logic MTL, thus sound for all left-continuous t-norms.

Propositional t-norm logics can be extended to their first-order and higher-order variants. These are needed for mathematical reasoning about fuzzy properties and will be employed later in this paper. For the formal apparatus of first-order fuzzy logic I refer the reader to [2]; Higher-order fuzzy logic has been introduced in [4] and described in a primer [5] freely available online. Here we shall only recall that the quantifiers $\forall, \exists$ are respectively realized as the infimum and supremum of the truth values, and that higher-order logic is a theory of fuzzy sets and relations with terms $\{x \mid \varphi(x)\}$, each of which represents the fuzzy set to which any element x belongs to the degree given by the truth value of the formula $\varphi(x)$.

3. Fuzzy logics as logics of costs

In fuzzy logic, truth values $x \in [0, 1]$ are usually interpreted as degrees of truth, with 1 representing full truth and 0 full falsity of a proposition. As argued in [1], the truth values can also be interpreted as measuring *costs,* with propositional connectives representing natural operations on costs. Under this interpretation, we abstract from the nature of costs (be they time, money, space, or any other kind of resources) and only assume that they are linearly ordered and normalized into the interval $[0, 1]$.

(The assumption of linear ordering can actually be relaxed to more general *prelinear* orderings, which cover most usual kinds of resources. In particular, direct products of linear orderings fall within the class, which allows *vectors* of costs, e.g., pairs of disk space and computation time, to be represented within this framework. In general, the cost-interpretation of fuzzy logic is based on the fact that most common resources show the structure of a prelinear residuated lattice. However, for simplicity we shall only consider linearly ordered costs that can be embedded in the real unit interval here.)

Under the cost-based interpretation, the truth value 1 represents the zero cost ("for free") and the truth value 0 represents a maximal or unaffordable cost. Intermediary truth values represent various degrees of costliness, with the usual ordering of $[0, 1]$ inverse to that of costs (the truth values can thus be understood as expressing degrees of truth of the fuzzy predicate "is cheap"). Strong conjunction & represents the *fusion* of resources, or the "sum" of costs. Various left-continuous t-norms

Institucionální repozitář AV ČR http://hdl.handle.net/11104/0163282

represent various ways by which costs may sum, and particular t-norm logics thus capture the rules that govern particular ways of cost addition. For example, the Łukasiewicz t-norm $*_Ł$ corresponds to the *bounded sum* of costs: assume that costs sum up to a bound $b > 0$; if we normalize the interval $[0, b]$ to $[0, 1]$ with the cost $c \in [0, b]$ represented by $1 - c/b \in [0, 1]$, then the bounded sum on $[0, b]$ corresponds to the Łukasiewicz t-norm on $[0, 1]$, since

$$(1 - x) *_Ł (1 - y) = 1 - (x + y)$$

unless the bound 0 (representing b) is achieved. Similarly the product t-norm corresponds to the *unbounded sum* of costs (via the negative logarithm), with 0 representing the infinite cost. The Gödel t-norm corresponds to taking the *maximum* cost as the "sum", which is also natural for some kinds of costs (e.g., the disk space for temporary results of calculation, which can be erased before the program proceeds). Other left-continuous t-norms correspond to variously distorted addition of costs, possibly suitable under some rare circumstances.

Obviously, disjunction and weak conjunction correspond, respectively, to the minimum and maximum of the two costs. The meaning of implication is that of surcharge: the cost expressed by $A \to B$ is the cost needed for B, provided we have already got the cost of A. (Observe that if the cost of B is less than or equal to that of A, then indeed $A \to B$ evaluates to 1, as we have already got the cost of B if we have the cost of A; i.e., the "upgrade" from A to B is "for free", which is represented by the value 1.) The equivalence connective represents the "difference" (in terms of &) between two costs, and negation the remainder to the maximal cost.

Tautologies of a given t-norm logic represent combinations of costs that are always "for free". More importantly, tautologies of the form $A_1 \,\&\ldots\&\, A_n \to B$ express the rules of preservation of "cheapness", as their cost-based interpretation reads: if we have the costs of all A_i together, then we also have the cost of B. Particular t-norm fuzzy logics thus express the rules of reasoning *salvis expensis,* in a similar manner as classical Boolean tautologies of the above form express the rules of reasoning *salva veritate.*

In the following sections we shall apply this interpretation of fuzzy logic to a particular kind of costs, namely the costs of program runs as modeled in propositional dynamic logic.

4. Propositional dynamic logic

Propositional dynamic logic (PDL) provides an abstract apparatus for logical modeling of behavior of programs. For details on PDL see [6, 7].

PDL models programs as (non-deterministic) transitions in an abstract space of states. (As such, PDL programs can represent any kind of actions over an arbitrary set of states, not only programs operating on the states of a computer; the applicability of both PDL and the present approach is thus much broader than just to computer programs.) Programs can in PDL be composed of simpler programs by means of a fixed set of constructions (the usual choice is that of regular expressions with tests, by which all common programming constructions are expressible), applied recursively on a fixed countable set of atomic programs (representing, e.g., the instructions of a processor). Propositional formulae of PDL express Boolean propositions about the states of the state space, and include, besides usual connectives of Boolean logic, modalities corresponding to programs, by means of which it is possible to reason about programs and their preconditions and postconditions.

Formally, the sets $\mathbf{Form}$ of formulae and $\mathbf{Prog}$ of programs of PDL are defined by simultaneous recursion from fixed countable sets of atomic formulae and atomic programs as follows:

- Every atomic formula is a formula; every atomic program is a program.
- If φ and ψ are formulae, then $\neg\varphi$ and $(\varphi \wedge \psi)$ are formulae (meaning *not* φ resp. φ *and* ψ). The abbreviations $\top$, $\bot$, $(\varphi \vee \psi)$, $(\varphi \to \psi)$, and $(\varphi \leftrightarrow \psi)$ are defined as usual in Boolean logic, with usual conventions on omitting parentheses.
- If α and β are programs, then α^*, $(\alpha \cup \beta)$, and $(\alpha; \beta)$ are programs (meaning *repeat* α *finitely many times, do* α *or* β, and *do* α *and then* β, respectively, where *or* and *finitely many* means a non-deterministic choice).
- If φ is a formula and α is a program, then $[\alpha]\varphi$ is a formula (meaning φ *holds after any run of* α). The expression $\langle\alpha\rangle\varphi$ abbreviates $\neg[\alpha]\neg\varphi$.
- If φ is a formula, then $\varphi?$ is a program (meaning *continue iff* φ).

The semantic models of PDL are multimodal Kripke structures $\langle W, R, V\rangle$ with W a non-empty set (of states),

Institucionální repozitář AV ČR http://hdl.handle.net/11104/0163282

$R\colon \mathbf{Prog} \to 2^{W^2}$ an evaluation of programs by binary relations on W (representing possible transitions between states by the program), and $V\colon \mathbf{Form} \to 2^W$ an evaluation of formulae by subsets of W (namely, the sets of verifying states), such that

$$V_{\neg\varphi} = W \setminus V_\varphi \tag{1}$$
$$V_{\varphi\wedge\psi} = V_\varphi \cap V_\psi \tag{2}$$
$$V_{\langle\alpha\rangle\varphi} = R_\alpha \leftarrow V_\varphi \tag{3}$$
$$R_{\alpha;\beta} = R_\alpha \circ R_\beta \tag{4}$$
$$R_{\alpha\cup\beta} = R_\alpha \cup R_\beta \tag{5}$$
$$R_{\alpha^*} = R_\alpha^* \tag{6}$$
$$R_{\varphi?} = \mathrm{Id} \cap V_\varphi \tag{7}$$

where $\circ$ denotes the composition of relations, $\leftarrow$ the preimage under a relation, R^* the reflexive and transitive closure of R, and Id the identity of relations. A formula φ is valid in the model iff $V_\varphi = W$, and is a tautology iff is valid in all models.

PDL is sound and complete w.r.t. the axiomatic system consisting of all propositional tautologies, the axioms

$$[\alpha;\beta]\varphi \leftrightarrow [\alpha][\beta]\varphi \tag{8}$$
$$[\alpha\cup\beta]\varphi \leftrightarrow [\alpha]\varphi \wedge [\beta]\varphi \tag{9}$$
$$[\alpha^*]\varphi \leftrightarrow \varphi \wedge [\alpha][\alpha^*]\varphi \tag{10}$$
$$[\varphi?]\psi \leftrightarrow (\varphi \to \psi) \tag{11}$$
$$[\alpha](\varphi \to \psi) \to ([\alpha]\varphi \to [\alpha]\psi) \tag{12}$$

and the rules of modus ponens (from φ and $\varphi \to \psi$ infer ψ), necessitation (from φ infer $[\alpha]\varphi$), and induction (from $\varphi \to [\alpha]\varphi$ infer $\varphi \to [\alpha^*]\varphi$).

For simplicity, we shall not consider expansions of PDL by further program constructions like intersection, converse, etc.

5. Modeling the costs of program runs

PDL does not take costs of program runs into consideration. In PDL, possible runs of a program α are modeled as transitions from a state w to a state w' such that $R_\alpha ww'$. The relation R_α representing the program α is binary (crisp): thus the states w' are either accessible or unaccessible from w by a run of α. In practice, however, it often occurs that although a state w' is theoretically achievable from w by α, the run of α from w to w' is not *feasible*—e.g., is too long (for example, needs to perform 10^{100} instructions, a frequent case in exponentially complex problems), requires too much memory, etc. Obviously, such unfeasible runs should not play a role in the practical assessment whether some condition φ can or cannot hold after the possible runs of α. Nevertheless, classical PDL cannot exclude such unfeasible runs, as there is no sharp boundary between feasible and unfeasible runs (i.e., the feasibility of runs is a fuzzy property).

A more realistic model can be obtained by considering costs of program runs, by means of which we can measure their feasibility. A simple model, which nevertheless covers many common situations, would assign the triple α, w, w' such that $R_\alpha ww'$ in a model of PDL a real number $C_\alpha ww'$ representing the cost of the run of α from w to w'. The cost thus would be represented by a function

$$C\colon \mathbf{Prog} \times W^2 \to [0, +\infty],$$

i.e., we are weighting the arrows in the co-graph of R_α by their costs; we assign the cost $+\infty$ to impossible runs with $\neg R_\alpha ww'$. The cost of a run of $\alpha_1; \alpha_2; \ldots; \alpha_n$ going from w_0 through $w_1, w_2, \ldots$ to w_n would be a function f (most often, the sum) of the costs of the runs of α_i from w_{i-1} to w_1. If there are different paths between w_0 and w_n through which $\alpha_1; \alpha_2; \ldots; \alpha_n$ can run, we are interested in the cheapest path, i.e., the run of $\alpha; \beta$ from w to w' will be understood as costing

$$C_{\alpha;\beta} ww' = \inf_{w''} f(C_\alpha ww'', C_\beta w''w'). \tag{13}$$

This model would allow us to work with the costs of program runs in the expanded models of PDL and define and investigate many useful notions related to costs by means of classical mathematics and logic. Nevertheless, since the important property of *feasibility* of a program run is essentially a fuzzy predicate, we shall recast this model in terms of the cost-based interpretation of fuzzy logic. This will allow us to employ fuzzy logic for a convenient definition of feasible runs and use the apparatus of fuzzy logic for reasoning about the costs on the propositional level, by replacing classical rules of reasoning with those of fuzzy logic. For a methodological discussion of this approach see [4, 5, 8, 9].

Thus we shall assume that the structure of costs is that of some t-norm algebra (see Section 3 for possible extension to more general algebras). Then, instead of weighting the arrows in the co-graph of R_α with costs, we can directly replace R_α with a *fuzzy relation* $\tilde{R}_\alpha \in [0,1]^{W^2}$, with the truth values of $\tilde{R}_\alpha ww'$ representing the cost of the run of α from w to w'.

Since the sum of costs now translates to conjunction in a suitable t-norm logic and since we are interested in the cheapest runs if more paths are possible, (13) now

Institucionální repozitář AV ČR http://hdl.handle.net/11104/0163282

translates to

$$\tilde{R}_{\alpha;\beta}ww' \equiv (\exists w'')(\tilde{R}_\alpha ww'' \mathbin{\&} \tilde{R}_\beta w''w') \tag{14}$$

with logical symbols interpreted in a t-norm fuzzy logic, i.e., in semantics,

$$\tilde{R}_{\alpha;\beta}ww' = \sup_{w''}(\tilde{R}_\alpha ww'' * \tilde{R}_\beta w''w')$$

It can be observed that the formula (14) has exactly the same form as in classical PDL where $R_{\alpha;\beta} = R_\alpha \circ R_\beta$, since by definition

$$(R_\alpha \circ R_\beta)ww' \equiv (\exists w'')(R_\alpha ww'' \mathbin{\&} R_\beta w''w') \tag{15}$$

The only difference between (14) and (15) is that the relations in (14) are fuzzy, and that the logical operations are (therefore) interpreted in a t-norm fuzzy logic instead of Boolean logic. This is in fact a general feature of using the framework of formal fuzzy logic that natural definitions usually take the same form as in the crisp case, only with the logical symbols reinterpreted in fuzzy logic (cf. [4, 5, 8, 9]): we shall see that further definitions will follow this pattern, too. Indeed, analogously to (15) it is usual [10] in fuzzy logic to define the composition of fuzzy relations $\tilde{R}$ and $\tilde{S}$ as

$$(\tilde{R} \circ \tilde{S})ww' \equiv (\exists w'')(\tilde{R}ww'' \mathbin{\&} \tilde{S}w''w'), \text{ i.e.,}$$
$$\equiv \sup_{w''}(\tilde{R}ww'' * \tilde{S}w''w')$$

Consequently, we can write

$$\tilde{R}_{\alpha;\beta} = \tilde{R}_\alpha \circ \tilde{R}_\beta$$

in our setting, in full analogy with the definition (4) of $R_{\alpha;\beta}$ in classical PDL.

Similarly it is natural to assume $\tilde{R}_{\alpha\cup\beta} = \tilde{R}_\alpha \cup \tilde{R}_\beta$ as in (5), where $(\tilde{R} \cup \tilde{S})ww'$ is defined for any fuzzy relations $\tilde{R}, \tilde{S}$ as $\tilde{R}ww' \vee \tilde{S}ww'$, since the cost of a run of $\alpha \cup \beta$ between w and w' should be the smaller of the two costs of the runs of α and β between the same states (which in $[0,1]_*$ is represented by the larger of the two truth values). Analogously one verifies that the cost of α^* is represented by the transitive and reflexive closure $\tilde{R}^*_\alpha$ of the fuzzy relation $\tilde{R}_\alpha$ defined as usual in the theory of fuzzy relations [10], in full analogy to (6).

The reinterpretation in fuzzy logic of (3), which expands to

$$V_{\langle\alpha\rangle\varphi}w \equiv (\exists w')(R_\alpha ww' \mathbin{\&} V_\varphi w') \tag{16}$$

yields a very natural modality expressing that after a *feasible* run of α the condition φ can hold. (Notice that this definition reflects the motivation for taking the costs of program runs into account, described in the beginning of this section.)

It can be observed in (16) that even if V_φ is crisp, a fuzzy R_α will yield a fuzzy $V_{\langle\alpha\rangle\varphi}$. Thus, because of the interplay of programs and formulae in PDL, our fuzzification of programs necessitates a fuzzification of formulae as well. A model of our fuzzified PDL is thus a triple $\langle W, \tilde{R}, \tilde{V}\rangle$, where W is a non-empty crisp set of states, $\tilde{R}$ maps programs α to fuzzy relations $\tilde{R}_\alpha \in [0,1]^{W^2}$, and $\tilde{V}$ gives fuzzy sets $\tilde{V}_\varphi \in [0,1]^W$ of states which fuzzily validate φ (i.e., $\tilde{V}_\varphi w$ is the truth value of φ in w).

Thus in the fuzzified (16), which reads

$$\tilde{V}_{\langle\alpha\rangle\varphi}w \equiv (\exists w')(\tilde{R}_\alpha ww' \mathbin{\&} \tilde{V}_\varphi w'), \tag{17}$$

the subformula $\tilde{R}_\alpha ww'$ can be understood as expressing the fuzzy proposition "w' is cheaply accessible from w by a run of α" (which is a fuzzy-propositional reading of the cost represented by $\tilde{R}_\alpha ww'$) and $\tilde{V}_\varphi w'$ as the fuzzy proposition "φ holds in w'" (viz, to the degree expressed by $\tilde{V}_\varphi w'$). Both $\tilde{R}_\alpha ww'$ and $\tilde{V}_\varphi w'$ can thus be understood as fuzzy propositions, and their combination in a single formula thus does not present a type mismatch: we only assume that the cost is represented by a truth value of the fuzzy proposition "the run is cheap", and that the mapping of costs to $[0,1]_*$ is such that the conjunction $*$ of truth values coincides with summation of costs. (This assumption is more natural if $\tilde{V}_\varphi$ for non-modal φ are assumed to be crisp, since then the fuzziness of $\tilde{V}_\psi$ for modal ψ arise exactly from considering the costs $\tilde{R}_\alpha ww'$ in (16). However, in many real-world applications of fuzzified PDL it may be desirable to have non-modal formulae fuzzy as well: then, if different algebras of degrees are needed for $\tilde{V}$ and $\tilde{R}$ in a particular model, one can use suitable direct products of t-norm algebras; I omit details here.) Particular interpretations $*$ of & and particular mappings of actual costs under consideration to $[0,1]_*$ will then yield concrete ways of calculating the truth values of this expression in particular models; importantly, however, the rules of general fuzzy logics like BL or MTL allow deriving theorems on program costs that are valid independently of a concrete representation in $[0,1]_*$.

Returning to (16), one can observe that again it coincides with the usual definition of preimage of a fuzzy set in a fuzzy relation (see, e.g., [11]). Thus we can write

$$\tilde{V}_{\langle\alpha\rangle\varphi} = \tilde{R}_\alpha \leftarrow \tilde{V}_\varphi,$$

again in full analogy with (3).

The derived semantical clause for $[\alpha]\varphi$, which in the classical case reads

$$V_{[\alpha]\varphi}w \equiv (\forall w')(R_\alpha ww' \rightarrow V_\varphi w'), \tag{18}$$

Institucionální repozitář AV ČR http://hdl.handle.net/11104/0163282

yields in the fuzzy reinterpretation

$$\tilde{V}_{[\alpha]\varphi} w \equiv (\forall w')(\tilde{R}_\alpha ww' \to \tilde{V}_\varphi w'), \tag{19}$$

a useful fuzzy modality expressing that after all feasible (or cheap enough) runs of α the fuzzy condition φ will hold. (Similar comments as in the case of $\langle\alpha\rangle\varphi$ are applicable.) The operation defined by (18) for crisp R_α and V_α and by (19) for fuzzy $\tilde{R}_\alpha$ and $\tilde{V}_\alpha$ is denoted by $\overset{\leftarrow}{\triangleleft}$ and called the *subproduct preimage* in [11], where it is studied as a particular case of BK-subproduct $\triangleleft$. (These notions were introduced by Bandler and Kohout in [12] for crisp relations and generalized for fuzzy relations in [13]. Further references to the literature on $\overset{\leftarrow}{\triangleleft}$ and its properties in fuzzy logic are given in [11].) Thus we can write

$$V_{[\alpha]\varphi} = R_\alpha \overset{\leftarrow}{\triangleleft} V_\varphi$$
$$\tilde{V}_{[\alpha]\varphi} = \tilde{R}_\alpha \overset{\leftarrow}{\triangleleft} \tilde{V}_\varphi$$

respectively for crisp and fuzzy PDL. Notice that unlike in classical PDL, $[\alpha]\varphi$ and $\langle\alpha\rangle\varphi$ are no longer interdefinable in fuzzified PDL, as the clauses (17) and (19) do not generally satisfy $\tilde{V}_{\neg\langle\alpha\rangle\varphi} = \tilde{V}_{[\alpha]\neg\varphi}$ in fuzzy logic, unless the negation $\neg$ is involutive. Both $[\alpha]$ and $\langle\alpha\rangle$ therefore need to be present in the language of fuzzified PDL as primitive symbols.

As an example of theorems that can be proved in our framework, we shall check the soundness of the axioms (8)–(12) and the three inference rules of classical PDL in our fuzzified PDL semantics. The validity of the axiom (8) in any model $M = \langle W, \tilde{R}, \tilde{V}\rangle$ is proved as follows:

$$\begin{aligned} & M \models [\alpha;\beta]\varphi \leftrightarrow [\alpha][\beta]\varphi \\ \text{iff} \quad & \tilde{V}_{[\alpha;\beta]\varphi} = \tilde{V}_{[\alpha][\beta]\varphi} \\ \text{iff} \quad & \tilde{R}_{\alpha;\beta} \overset{\leftarrow}{\triangleleft} \tilde{V}_\varphi = \tilde{R}_\alpha \overset{\leftarrow}{\triangleleft} \tilde{V}_{[\beta]\varphi} \\ \text{iff} \quad & (\tilde{R}_\alpha \circ \tilde{R}_\beta) \overset{\leftarrow}{\triangleleft} \tilde{V}_\varphi = \tilde{R}_\alpha \overset{\leftarrow}{\triangleleft} (\tilde{R}_\beta \overset{\leftarrow}{\triangleleft} \tilde{V}_\varphi), \end{aligned}$$

where the last identity is an easy property of $\overset{\leftarrow}{\triangleleft}$, see [11, Cor. 5.17].

Similarly, the validity of the axiom (9) is proved by

$$\begin{aligned} & M \models [\alpha\cup\beta]\varphi \leftrightarrow [\alpha]\varphi \wedge [\beta]\varphi \\ \text{iff} \quad & \tilde{V}_{[\alpha\cup\beta]\varphi} = \tilde{V}_{[\alpha]\varphi\wedge[\beta]\varphi} \\ \text{iff} \quad & \tilde{R}_{\alpha\cup\beta} \overset{\leftarrow}{\triangleleft} \tilde{V}_\varphi = \tilde{V}_{[\alpha]\varphi} \cap_\wedge \tilde{V}_{[\alpha]\varphi} \\ \text{iff} \quad & (\tilde{R}_\alpha \cup \tilde{R}_\beta) \overset{\leftarrow}{\triangleleft} \tilde{V}_\varphi = (\tilde{R}_\alpha \overset{\leftarrow}{\triangleleft} \tilde{V}_\varphi) \cap_\wedge (\tilde{R}_\beta \overset{\leftarrow}{\triangleleft} \tilde{V}_\varphi), \end{aligned}$$

where the last identity is again an easy property of $\overset{\leftarrow}{\triangleleft}$, see [11, Cor. 5.16]. Notice that weak conjunction $\wedge$ is in order in the fuzzy version of (9), corresponding in the proof to *min-intersection* defined for any fuzzy sets $\tilde{U}, \tilde{V}$ as $(\tilde{U} \cap_\wedge \tilde{V})w \equiv \tilde{U}w \wedge \tilde{V}w$.

In order to verify the axiom (10), we need a few definitions and lemmata. First, define for any fuzzy relation $\tilde{R}$ its iterations

$$\tilde{R}^0 = \mathrm{Id} \tag{20}$$
$$\tilde{R}^{n+1} = \tilde{R} \circ \tilde{R}^n \tag{21}$$

for all $n \in \mathbb{N}$. Furthermore, the union $\bigcup \mathcal{A}$ of a crisp or fuzzy set $\mathcal{A}$ of fuzzy relations is in higher-order fuzzy logic defined as

$$(\bigcup \mathcal{A})ww' \equiv (\exists \tilde{R})(\mathcal{A}\tilde{R} \;\&\; \tilde{R}ww').$$

Obviously, for any fuzzy relation $\tilde{R}$,

$$\bigcup_{n=0}^{\infty} \tilde{R}^n = \tilde{R}^0 \cup \bigcup_{n=1}^{\infty} \tilde{R}^n = \mathrm{Id} \cup \bigcup_{n=1}^{\infty} \tilde{R}^n$$

by (20). It can trivially be verified that by definitions, $\mathrm{Id} \overset{\leftarrow}{\triangleleft} \tilde{V} = \tilde{V}$, thus also $\tilde{R}^0 \overset{\leftarrow}{\triangleleft} \tilde{V} = \tilde{V}$, for any fuzzy relation $\tilde{R}$ and any fuzzy set $\tilde{V}$. Finally, it can be proved (cf. [10]) that the transitive and reflexive closure $\tilde{R}^*$ of a fuzzy relation $\tilde{R}$ is in fuzzy logic characterized in the same way as in classical mathematics, viz

$$\tilde{R}^* = \bigcup_{n=0}^{\infty} \tilde{R}^n = \mathrm{Id} \cup \bigcup_{n=1}^{\infty} \tilde{R}^n$$

Now we can show the soundness of (10), which amounts to the general validity of $\tilde{V}_{[\alpha^*]} = \tilde{V}_{\varphi\wedge[\alpha][\alpha^*]\varphi}$. We have the following chain of identities, justified by definitions and previous lemmata:

$$\begin{aligned} \tilde{V}_{[\alpha^*]\varphi} &= \tilde{R}_{\alpha^*} \overset{\leftarrow}{\triangleleft} \tilde{V}_\varphi = \\ &= \Big(\bigcup_{n=0}^{\infty} \tilde{R}_\alpha^n\Big) \overset{\leftarrow}{\triangleleft} \tilde{V}_\varphi \\ &= \Big(\mathrm{Id} \cup \bigcup_{n=1}^{\infty} \tilde{R}_\alpha^n\Big) \overset{\leftarrow}{\triangleleft} \tilde{V}_\varphi \\ &= (\mathrm{Id} \overset{\leftarrow}{\triangleleft} \tilde{V}_\varphi) \cap_\wedge \Big(\Big(\bigcup_{n=1}^{\infty} \tilde{R}_\alpha^n\Big) \overset{\leftarrow}{\triangleleft} \tilde{V}_\varphi\Big) \\ &= \tilde{V}_\varphi \cap_\wedge \Big(\Big(\tilde{R}_\alpha \circ \bigcup_{n=0}^{\infty} \tilde{R}_\alpha^n\Big) \overset{\leftarrow}{\triangleleft} \tilde{V}_\varphi\Big) \\ &= \tilde{V}_\varphi \cap_\wedge \tilde{V}_{[\alpha;\alpha^*]\varphi} = \tilde{V}_{\varphi\wedge[\alpha][\alpha^*]\varphi}. \end{aligned}$$

Notice again that weak conjunction is in order in fuzzified (10).

The soundness of the rule of induction amounts to the validity of inferring

$$\tilde{V}_\varphi \subseteq \tilde{R}_\alpha^* \overset{\leftarrow}{\triangleleft} \tilde{V}_\varphi \quad \text{from} \quad \tilde{V}_\varphi \subseteq \tilde{R}_\alpha \overset{\leftarrow}{\triangleleft} \tilde{V}_\varphi.$$

Institucionální repozitář AV ČR http://hdl.handle.net/11104/0163282

By induction, we shall prove that from $\tilde{V}_\varphi \subseteq \tilde{R}_\alpha \hookleftarrow \tilde{V}_\varphi$ we can infer $\tilde{V}_\varphi \subseteq \tilde{R}^n_\alpha \hookleftarrow \tilde{V}_\varphi$ for all $n \in \mathbb{N}$, i.e., by [14, Lemma B.8(L5)],

$$\tilde{V}_\varphi \subseteq \bigcap_{n\in\mathbb{N}} (\tilde{R}^n_\alpha \hookleftarrow \tilde{V}_\varphi),$$

which is by [11, Cor. 5.16] equivalent to the required

$$\tilde{V}_\varphi \subseteq \Big(\bigcup_{n\in\mathbb{N}} \tilde{R}^n_\alpha\Big) \hookleftarrow \tilde{V}_\varphi.$$

The first step $\tilde{V}_\varphi \subseteq \tilde{R}^0_\alpha \hookleftarrow \tilde{V}_\varphi$ of the induction is trivially valid by $\tilde{R}^0_\alpha \hookleftarrow \tilde{V}_\varphi = \mathrm{Id} \hookleftarrow \tilde{V}_\varphi = \tilde{V}_\varphi$. For the induction step, we need to infer

$$\tilde{V}_\varphi \subseteq \tilde{R}^{n+1}_\alpha \hookleftarrow \tilde{V}_\varphi \quad \text{from} \quad \tilde{V}_\varphi \subseteq \tilde{R}^n_\alpha \hookleftarrow \tilde{V}_\varphi,$$

i.e., by [14, Th. 4.3(I14)],

$$(\tilde{R}^n_\alpha \circ \tilde{R}_\alpha) \rightharpoonup \tilde{V}_\varphi \subseteq \tilde{V}_\varphi, \quad \text{from} \quad \tilde{R}_\alpha \rightharpoonup \tilde{V}_\varphi \subseteq \tilde{V}_\varphi.$$

By [11, Cor. 4.14], the former is equivalent to

$$\tilde{R}_\alpha \rightharpoonup (\tilde{R}^n_\alpha \rightharpoonup \tilde{V}_\varphi) \subseteq \tilde{V}_\varphi,$$

which follows from $\tilde{R}_\alpha \rightharpoonup \tilde{V}_\varphi \subseteq \tilde{V}_\varphi$ by monotony of $\rightharpoonup$ w.r.t. $\subseteq$ [11, Cor. 4.7].

A discussion of the test construction is postponed to Section 6; therefore we shall skip checking the soundness of the the axiom (11). The soundness of the rule of modus ponens and the axioms of propositional logic is demonstrated in [15], as $\langle W, \tilde{V}\rangle$ forms the usual intensional semantics for fuzzy logic. The soundness of the rule of necessitation amounts to the validity of inferring $W \subseteq \tilde{R}_\alpha \hookleftarrow \tilde{V}_\varphi$, i.e., $\tilde{R}_\alpha \rightharpoonup W \subseteq \tilde{V}_\varphi$, from $W \subseteq \tilde{V}_\varphi$; but since $\tilde{R}_\alpha$ only operates on W, it is immediate that $\tilde{R}_\alpha \rightharpoonup W \subseteq W \subseteq \tilde{V}_\varphi$.

On the other hand, the axiom (12) fails in fuzzy PDL, as it is well known (already from [2]) that fuzzified Kripke frames do not in general validate the modal axiom K. Since also the interdefinability of $\langle\alpha\rangle$ and $[\alpha]$ fails for non-involutive negation, dual axioms and rules for $\langle\alpha\rangle$ need to be added to a prospective axiomatic system of fuzzified PDL. I omit the discussion of these axioms here; it can nevertheless be hinted that since the relationship between the semantic clauses for $\langle\alpha\rangle$ and $[\alpha]$ is that of Morsi's duality [16] (combined with the duality between fuzzy relations and their converses), the formulation and soundness of the dual axioms and rules for $\langle\alpha\rangle$ can be obtained from the axioms and rules for $[\alpha]$ automatically by the same duality.

6. The role of tests

In classical PDL, tests $\varphi?$ have the role in branching complex programs: they are employed in definitions of such programming constructions as if–then–else, while–do, or repeat–until. They do not themselves affect the state in which a program run is, but bar a further execution of the program if their condition is not met. A straightforward fuzzification of the semantic condition (7), $\tilde{R}_{\varphi?} = \mathrm{Id} \cap \tilde{V}_\varphi$, would interpret tests in fuzzy PDL as programs which do not change the state, but can decrease the "passability" of the run through the current state according to the truth value of the condition φ. This, however, does not correspond to the primary motivation of $\tilde{R}_\alpha ww'$ as the *cost* of the run of α from w to w': the condition φ may be cheap to test, but can have a low truth degree in w, or vice versa. The two roles of the truth value yielded by the test $\varphi?$ do not match in fuzzy PDL: the *truth degree* of φ should affect the possibility of further execution, while the *cost* of performing the test of φ should contribute to the overall cost of the run of a complex program. Neither of the two roles can be sacrificed, since the former is necessary for branching the program (by the fuzzy if–then–else and cycle constructions), while without the latter we would be unable to distinguish between feasible and unfeasible runs (which was our primary motivation for the fuzzification of PDL).
Unless we want to stipulate that the conventional complexity (or cost) of a test be identified with the truth value it yields, thus equating the accessibility of paths of program execution with their costs (by which the actual cost of performing the computation is replaced by a different conventional measure), we may have to admit that the identification of the feasibility (or cost) value with the value of accessibility was too bold and that these two fuzzy relations on W have to be distinguished. If we denote the fuzzy accessibility relation by $\tilde{R}_\alpha$ and the feasibility relation by $\tilde{C}_\alpha$, then the test $\varphi?$ would contribute to $\tilde{R}_\alpha$ by the truth value of φ, while to $\tilde{C}_\alpha$ by the cost of performing the test. For instance, performing a test of a difficult tautology may contribute a lot to the cost of the run, while not decreasing the "correctness" degree of the run at all. We may then distinguish the modality $\langle\alpha\rangle^{\tilde{R}}\varphi$ expressing that there is a "correct" run to a state where φ holds from $\langle\alpha\rangle^{\tilde{R}\cap\tilde{C}}\varphi$ expressing that there is a "correct feasible" run validating φ (all conditions understood fuzzily). Their semantic clauses are, respectively:

$$\tilde{V}_{\langle\alpha\rangle^{\tilde{R}}\varphi} w \equiv (\exists w')(\tilde{R}_\alpha ww' \mathbin{\&} \tilde{V}_\varphi w')$$
$$\tilde{V}_{\langle\alpha\rangle^{\tilde{R}\cap\tilde{C}}\varphi} w \equiv (\exists w')(\tilde{R}_\alpha ww' \mathbin{\&} \tilde{C}_\alpha ww' \mathbin{\&} \tilde{V}_\varphi w')$$

The apparatus of costs of program runs thus appears

Institucionální repozitář AV ČR http://hdl.handle.net/11104/0163282

to operate best on PDL with fuzzified accessibility relations of programs, whose truth degrees do not express the degrees of feasibility (or costs) of program runs, but the degrees of their admissibility (or "correctness", in the sense of the satisfaction of conditions passed through). The fuzzification of admissibility can be developed independently, without regarding costs of runs at all, thus making the same idealization as regards costs as classical PDF, i.e., with equating feasibility and admissibility of runs. Such fuzzification only generalizes the framework of PDL to permit fuzzy conditions like "if the temperature is high, do α" (which may be quite useful in real-world applications) and a measure of "correctness" of some transitions between states by programs (capturing for instance such phenomena as rounding numerical results etc.).

Adding moreover the apparatus for costs then makes the (already fuzzified) model more realistic by the possibility of distinguishing not only (the degree of) correctness, but also (the degree of) feasibility of (more or less correct) runs of programs. The double nature of tests regarding the truth and cost degrees, however, seems to exclude the possibility of adding the apparatus of costs directly to crisp rather than already fuzzified PDL, unless we forbid tests on feasibility (e.g., of the form $(\langle\alpha\rangle^{\tilde{R}\cap\tilde{C}}\varphi)?$), which automatically fuzzify the admissibility of runs.

Various kinds of restrictions on tests (e.g., allowing only tests of atomic formulae, non-modal formulae, formulae not referring to feasibility, etc.) would, however, strongly affect the requirements on the models and their properties. An elaboration of these considerations is left for future work, as are the problems of axiomatizability of such systems of fuzzy PDL and a detailed investigation of their properties.

References

[1] L. Běhounek, "Fuzzy logics interpreted as logics of resources," in *XXII Logica Volume of Abstracts*, (Prague), Institute of Philosophy, Academy of Sciences of the Czech Republic, 2008. XXII International Conference Logica, held on June 16–19, 2008 in Hejnice, Czech Republic.

[2] P. Hájek, *Metamathematics of Fuzzy Logic*, vol. 4 of *Trends in Logic*. Dordercht: Kluwer, 1998.

[3] F. Esteva and L. Godo, "Monoidal t-norm based logic: Towards a logic for left-continuous t-norms," *Fuzzy Sets and Systems*, vol. 124, no. 3, pp. 271–288, 2001.

[4] L. Běhounek and P. Cintula, "Fuzzy class theory," *Fuzzy Sets and Systems*, vol. 154, no. 1, pp. 34–55, 2005.

[5] L. Běhounek and P. Cintula, "Fuzzy Class Theory: A primer v1.0," Tech. Rep. V-939, Institute of Computer Science, Academy of Sciences of the Czech Republic, Prague, 2006. Available at www.cs.cas.cz/research/library/reports_900.shtml.

[6] D. Harel, "Dynamic logic," in *Handbook of Philosophical Logic* (D. M. Gabbay and F. Guenthner, eds.), vol. II: Extensions of Classical Logic, pp. 497–604, Dordrecht: D. Reidel, 1st ed., 1984.

[7] D. Harel, D. Kozen, and J. Tiurin, *Dynamic Logic*. Cambridge MA: MIT Press, 2000.

[8] L. Běhounek and P. Cintula, "From fuzzy logic to fuzzy mathematics: A methodological manifesto," *Fuzzy Sets and Systems*, vol. 157, no. 5, pp. 642–646, 2006.

[9] L. Běhounek and P. Cintula, "Fuzzy class theory as foundations for fuzzy mathematics," in *Fuzzy Logic, Soft Computing and Computational Intelligence: 11th IFSA World Congress*, vol. 2, (Beijing), pp. 1233–1238, Tsinghua University Press/Springer, 2005.

[10] L. A. Zadeh, "Similarity relations and fuzzy orderings," *Information Sciences*, vol. 3, pp. 177–200, 1971.

[11] L. Běhounek and M. Daňková, "Relational compositions in Fuzzy Class Theory." To appear in Fuzzy Sets and Systems (doi:10.1016/j.fss.2008.06.013), 2008.

[12] W. Bandler and L. J. Kohout, "Mathematical relations, their products and generalized morphisms," Tech. Rep. EES-MMS-REL 77-3, Man–Machine Systems Laboratory, Department of Electrical Engineering, University of Essex, Essex, Colchester, 1977.

[13] W. Bandler and L. J. Kohout, "Fuzzy relational products and fuzzy implication operators," in *International Workshop of Fuzzy Reasoning Theory and Applications*, (London), Queen Mary College, University of London, 1978.

[14] L. Běhounek, U. Bodenhofer, and P. Cintula, "Relations in Fuzzy Class Theory: Initial steps," *Fuzzy Sets and Systems*, vol. 159, no. 14, pp. 1729–1772, 2008.

[15] L. Běhounek, "Fuzzification of Groenendijk–Stokhof propositional erotetic logic," *Logique et Analyse*, vol. 47, no. 185–188, pp. 167–188, 2004.

[16] N. N. Morsi, W. Lotfallah, and M. El-Zekey, "The logic of tied implications, part 2: Syntax," *Fuzzy Sets and Systems*, vol. 157, pp. 2030–2057, 2006.

Institucionální repozitář AV ČR http://hdl.handle.net/11104/0163282