THE SET OF FORMULAS OF PrAL$^+$
VALID IN A FINITE STRUCTURE IS UNDECIDABLE

Anna Borowska
Faculty of Computer Science, Bialystok University of Technology, Bialystok, Poland

Abstract: We consider a probabilistic logic of programs. In [6] it is proved that the set of formulas of the logic PrAL, valid in a finite structure, is decidable with respect to the diagram of the structure. We add to the language $L_P$ of PrAL a sign $\bigcup$ and a functor $lg$. Next we justify that the set of formulas of extended logic, valid in a finite at least 2-element structure (for $L_P^+$) is undecidable.

Keywords: Probabilistic Algorithmic Logic, existential iteration quantifier

1. Introduction

In [6] the Probabilistic Algorithmic Logic PrAL is considered, constructed for expressing properties of probabilistic algorithms understood as iterative programs with two probabilistic constructions $x := \text{random}$ and $\text{either}_p \ldots \text{or} \ldots \text{ro}$. In order to describe probabilities of behaviours of programs a sort of variables (interpreted as real numbers) and symbols $+, -, *, 0, 1, <$ (interpreted in the standard way in the ordered field of real numbers) was added to the language $L_P$ of PrAL.

In the paper [5] the changes of information which depend on realizations of probabilistic program was considered. That’s why the language $L_P$ was extended by adding the sign $\bigcup$ (called the existential iteration quantifier) and the functor $lg$ (for the one-argument operation of a logarithm with a base 2 interpreted in the real ordered field). The new language was denoted by $L_P^+$.

The paper [6] contains an effective method of determining probabilities for probabilistic programs interpreted in a finite structure. The effectiveness of the method leads to the decidability of the set of formulas of $L_P^+$, valid in a fixed finite structure (provided that we have at our disposal a suitable finite part of the diagram of the structure). Here we shall justify that the set of probabilistic algorithmic formulas of $L_P^+$,
valid in an arbitrary, finite at least 2-element structure, is undecidable with respect to its diagram.

We shall start from a presentation of the syntax and the semantics of the language $L_P^+$. We use the syntax and the semantics of $L_P$ proposed by W. Danko in [6].

2. Syntax and Semantics of $L_P^+$

A language $L_P$ is an extension of a first-order language $L$ and includes three kinds of well-formed expressions: terms, formulas and programs. As mentioned above, the alphabet of $L_P^+$ contains two additional elements: the arithmetic one-argument functor $\lg$ and the sign $\bigcup$ (the existential iteration quantifier). An interpretation of $L_P^+$ relies on an interpretation of the first-order language $L$ in a structure $\mathfrak{I}$ (We take into consideration only finite structures. By finite structure we mean a structure with a finite, at least 2-element set $A$.) and on the standard interpretation of the language $L_{\mathbb{R}}$ in the ordered field of real numbers (cf. [6]).

The alphabet of the language $L_P^+$ contains

- a set of constants $C_P$, which consists of a finite subset $C = \{c_1, \ldots, c_u\}$ of symbols for each element of the set $A = \{a_1, \ldots, a_u\}$, a subset $C_{\mathbb{R}}$ of real constant symbols and a subset $C_L$ of logical constant symbols,
- an enumerable set $V_P = \{V \cup V_{R} \cup V_0\}$ of variables, where a subset $V = \{v_0, v_1, \ldots\}$ consists of non-arithmetic individual variables, a subset $V_{R} = \{x_0, x_1, \ldots\}$ contains real variables and a subset $V_0 = \{q_0, q_1, \ldots\}$ contains propositional variables,
- a set of signs of relations $\Psi_P = \{\Psi \cup \Psi_{R}\}$, where the subset $\Psi$ consists of non-arithmetic predicates and the subset $\Psi_{R} = \{<_{R}, =_{R}\}$ contains arithmetic predicates,
- an enumerable set of functors $\Phi_P = \{\Phi \cup \Phi_{R}\}$, which consists of the subset $\Phi_{R} = \{+_{R}, -, \cdot, \lg\}$ of symbols for arithmetic operations and the subset $\Phi$ of symbols for non-arithmetic operations,
- the set $\{\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow\}$ of logical connectives,
- the set $\{\text{if, then, else, fi, while, do, od, either, or, ro, random}\}$ of symbols for program constructions,
- the set $\{\exists, \forall\}$ of symbols for classical quantifiers (for real variables only),
- the existential iteration quantifier $\bigcup$.

1 For each probability distribution defined on a set $A$ we generate a different random assignment. We use a number $l$ to distinguish them.
The set of probabilistic algorithmic formulas of PrAL$^+$ valid in a finite structure is undecidable

- the set $\{(.)\}$ of auxiliary symbols.

In the language $L_P^+$, we distinguish two kinds of terms (arithmetic and non-arithmetic), formulas (classical and algorithmic) and programs.

The set of terms $T_P = \{ T \cup T_R \}$ of $L_P^+$ consists of a subset of non-arithmetic terms $T$ and a subset $T_R$ of arithmetic terms.

**Definition 2.1** The set $T$ of non-arithmetic terms is defined as the smallest set of expressions satisfying the following conditions:

- each constant of $C$ and each variable of $V$ belongs to $T$,
- if $\phi_i \in \Phi$ ($\phi_i$ - an $n_i$-argument functor ($n_i \geq 0$)) and $\tau_1, \ldots, \tau_{n_i} \in T$ then an expression $\phi_i(\tau_1, \ldots, \tau_{n_i})$ belongs to $T$.

**Definition 2.2** The set $T_R$ of arithmetic terms is the smallest set such that:

- each constant of $C_R$ and each real variable of $V_R$ belongs to $T_R$,
- if $t_1, t_2 \in T_R$ then expressions $t_1 + t_2$, $t_1 - t_2$, $t_1 \cdot t_2$, $\log t_1$ belong to $T_R$,
- if $\alpha$ is a formula of $L$ then $P(\alpha)$ belongs to $T_R$. (We read the symbol $P$ as follows "probability that").

**Definition 2.3** The set $F_O$ of open formulas is the smallest set such that:

- if $\tau_1, \ldots, \tau_{m_j} \in T$ and $\psi_j \in \Psi$ ($\psi_j$ - an $m_j$-argument predicate) then $\psi_j(\tau_1, \ldots, \tau_{m_j}) \in F_O$,
- if $\alpha, \beta \in F_O$ then expressions $\neg \alpha$, $\alpha \lor \beta$, $\alpha \land \beta$, $\alpha \Rightarrow \beta$, $\alpha \Leftrightarrow \beta$ belong to $F_O$.

**Definition 2.4** The set $\Pi$ of all programs is defined as the smallest set of expressions satisfying the following conditions:

- each expression of the form $v := \tau$ or $v := $random, where $v \in V$, $\tau \in T$ is a program,
- if $\gamma \in F_O$ and $M_1, M_2 \in \Pi$ then expressions $M_1; M_2$, if $\gamma$ then $M_1$ else $M_2$ fi, while $\gamma$ do $M_1$ od, either$\_p$ $M_1$ or $M_2$ ro ($p$ is a real number) are programs.

We establish that in an expression $\bigcup K \alpha$ (where $K$ is a program) the letter $\alpha$ denotes a formula which does not contain any iteration quantifiers.

**Definition 2.5** The set $F_P$ of all formulas of the language $L_P^+$ is the smallest extension of the set $F_O$ such that:

- if $t_1, t_2 \in T_R$ then $t_1 =_R t_2$, $t_1 <_R t_2$ belong to $F_P$,
- if $\alpha, \beta \in F_P$ then the expressions $-\alpha$, $\alpha \lor \beta$, $\alpha \land \beta$, $\alpha \Rightarrow \beta$, $\alpha \Leftrightarrow \beta$ belong to $F_P$. 

7
– if $\alpha \in F_P$ and $x \in V_\Re$ is a free variable in $\alpha$ then $\exists x \alpha$, $\forall x \alpha$ belong to $F_P$,

– if $K \in \Pi$ and $\alpha \in F_P$ then $K\alpha$ is a formula of $F_P$,

– if $K \in \Pi$ and $\alpha \in F_P$ then $\bigcup K \alpha$ belongs to $F_P$.

A variable $x$ is free in a formula $\alpha$ if $x$ is not bounded by any quantifier.

Let $L^+_P$ be a fixed algorithmic language of the type $\langle \{n_k\}_{k \in \Phi}, \{m_l\}_{l \in \Psi} \rangle$ where $\Phi$ and $\Psi$ are sets of fixed functions and relations, respectively.

We interpret non-arithmetic individual variables of $L^+_P$ as elements of $A$. Real variables are interpreted as elements of the set $\Re$ of real numbers.

Let’s denote the set of possible valuations $w$ of non-arithmetic variables by $W$.

**Definition 2.6** By the interpretation of a non-arithmetic term $\tau$ of $L_P$ in the structure $\Im$ we mean a function $\tau_\Im : W \mapsto A$ which is defined recursively.

– If $\tau$ is a variable $v \in V$ then $v_\Im (w) \equiv w(v)$.

– If $\tau$ is of the form $\phi(\tau_1, \ldots, \tau_n)$, where $\tau_1, \ldots, \tau_n \in T$ and $\phi \in \Phi$ is an $n$-argument functor then $\phi(\tau_1, \ldots, \tau_n)_\Im (w) \equiv \phi_\Im (\tau_1_\Im (w), \ldots, \tau_n_\Im (w))$, where $\tau_1_\Im (w), \ldots, \tau_n_\Im (w)$ are defined earlier.

To interpret random assignments (i.e. constructions of the form $v := \text{random}$) in a probabilistic way we assume that there exists a fixed probability distribution defined on $A$

$$\rho_i : A \mapsto [0, 1], \quad \sum_{i=1}^{\mu} \rho_i(a_i) = 1.$$  

**Definition 2.7** (cf. [6]) A pair $\langle \Im, \rho \rangle$, where $\rho$ is a set of fixed probability distributions $\rho_i$ defined on $A$ and $\Im$ is a structure for $L^+_P$, is called a probabilistic structure. In this structure we interpret probabilistic programs.

By $\mathcal{M}$ we denote the set of all probability distributions defined on the set $W$ of valuations of non-arithmetic variables such that

$$\mu : W \mapsto [0, 1], \quad \sum_{w \in W} \mu(w_i) \leq 1.$$  

By $S$ we mean the set of all states, i.e. all pairs $s = \langle \mu, w_\Re \rangle$, where $\mu$ is a probability distribution of valuations of non-arithmetic variables and $w_\Re$ is a valuation of real variables of $V_\Re$. 

8
**Definition 2.8** (cf. [6]) A probabilistic program $K$ is interpreted in the structure $< \mathcal{S}, \rho >$ as a partial function transforming the set of states into the set of states

$$K_{<\mathcal{S},\rho>} : S \mapsto S.$$  

Let $K(v_1, \ldots, v_h)$ represent a fixed program in $L_{P^+}$. An arbitrary program $K$ contains only a finite number of non-arithmetic variables. We denote this set of variables by $V = \{v_1, \ldots, v_h\}$. Since $A = \{a_1, \ldots, a_u\}$ is also a finite set, then a set of all possible valuations of program variables will be also finite. We denote it by $\{w_1, \ldots, w_n\}$, where $n = u^h$.

Let’s notice that programs do not operate on variables of $V \mathbb{R}$. Thus we can interpret an arbitrary program $K$ as partial functions transforming probability distributions defined on the set of valuations of program variables (cf. [6])

$$K_{<\mathcal{S},\rho>} : \mathcal{M} \mapsto \mathcal{M}.$$  

If $\mu$ is the input probability distribution of valuations of program variables (input probability distribution for short) then a realization of a program $K$ leads to a new output probability distribution $\mu'$ of valuations of program variables (output probability distribution for short). A distribution $\mu$ ($\mu'$) associates with each valuation $w$ of program variables a corresponding probability of its appearance.

The interpretation of program constructions (used in this paper) can be found in the Appendix.

An arithmetic term of the form $P(\alpha)$ denotes the probability, that the formula $\alpha$ of $L$ is satisfied at a distribution $\mu$ (cf. [6])

$$[P(\alpha)]_{\mathcal{S}}(s) = \sum_{w \in W} \mu(w), \text{ where } W^\alpha = \{w \in W : \mathcal{S}, w \models \alpha\}.$$  

Let $s = < \mu, w_\mathbb{R} >$ be a state and let $s' = < \mu', w_\mathbb{R} >$ represent the state $s' = K_{<\mathcal{S},\rho>}(s)$.

Given below is the interpretation of a formula $K\alpha$ ($\alpha \in F_\mathbb{P}$ and $K \in \Pi$).

$$(K\alpha)_{<\mathcal{S},\rho>} (s) = \begin{cases} \alpha_{<\mathcal{S},\rho>} (s') & \text{if } K_{<\mathcal{S},\rho>} (s) \text{ is defined and } s' = K_{<\mathcal{S},\rho>} (s) \\ \text{is not defined otherwise} & \end{cases}$$

The satisfiability of a formula $K\alpha$, where $\alpha \in F_\mathbb{P}$ and $K \in \Pi$, is defined in the following way (cf. [6])

$$< \mathcal{S}, \rho >, s \models K\alpha \text{ iff } < \mathcal{S}, \rho >, s' \models \alpha, \text{ where } s' = K_{<\mathcal{S},\rho>} (s).$$  

The next definition establishes the meaning of the existential iteration quantifier ($K \in \Pi, \alpha \in F_\mathbb{P}$).
We can informally express the formula $\bigcup K\alpha$ in the following way:

$$\alpha \lor K\alpha \lor K^2\alpha \lor \ldots$$

The satisfiability of a formula $\bigcup K\alpha~(K \in \Pi, \alpha \in F_P)$ is defined as an infinite alternative of formulas $(K^i\alpha)$ for $i \in N$.

**Example 2.10** Now we shall present a formula which contains the iteration quantifier. Let’s consider the formula $\beta$:

$$\begin{align*}
K_0: & \quad v_1 := 0; \\
K: & \quad \text{if } (v_1 = 0) \text{ then } v_1 := \text{random}_1; v_2 := 0; \text{ else } v_2 := 1; \text{ fi} \\
\alpha: & \quad x = P(v_1 = 1 \lor v_2 = 0)
\end{align*}$$

where $K_0$ and $K$ are programs interpreted in the structure $<\mathcal{I}, \rho>$ with a 2-element set $A = \{0, 1\}$. For a random assignment $v_1 := \text{random}_1$ we define the probability distribution $\rho_1 = [0.5, 0.5]$. The set of possible valuations of program variables contains 4 elements: $w_1 = (0, 0), w_2 = (0, 1), w_3 = (1, 0), w_4 = (1, 1)$. We carry out computations for the input probability distribution $\mu = [0.25, 0.25, 0.25, 0.25]$. $P(\gamma)$ denotes the probability that $\gamma$ is satisfied (at a distribution $\mu$). Let’s notice, that formula $\beta$ describes the following fact:

$$(x = 0) \lor (x = 0.5) \lor (x = 0.5 \ast 0.5) \lor (x = 0.5 \ast 0.5 \ast 0.5) \lor \ldots$$

3. The proof of the main lemma

As we have mentioned (it is proved in [6]), the set of probabilistic algorithmic formulas of PrAL valid in a finite structure for $L_P$ is decidable with respect to the diagram of the structure. By the diagram $D(\mathcal{I})$ of the structure $\mathcal{I}$ we understand the set of all atomic or negated atomic formulas $\phi(c_{i_1}, \ldots, c_{i_m}) = c_{i_0}$ ($\phi$ is a functor of $L$) and $\psi(c_{i_1}, \ldots, c_{i_m})$ ($\psi$ is a predicate symbol of $L$), which are valid in $\mathcal{I}$.

The proof of decidability of PrAL essentially uses the Lemma which reduces the problem of validity of sentences of $L_P$ to the (decidable) problem of the validity of sentences of the first-order arithmetic of real numbers. Finally, it appears that the set of formulas of PrAL, valid in all at most $u$-element structures for $L_P$, is decidable.

We shall show that if the language $L_P^+$ contains additionally the sign $\bigcup$ and the functor $\log$ (for the operation of a logarithm) we can define natural numbers and operations of addition and multiplication for natural numbers.

Let’s assume that $0.5^i$ abbreviates the expression $\overbrace{0.5 \ast 0.5 \ast \ldots \ast 0.5}^{i \text{ times}}$.  

10
Lemma 3.1 Let \(< \mathfrak{S}, \rho >\) be an arbitrary fixed probabilistic structure (for \(L^+ \rho\)) with a finite set \(A = \{a_1, a_2, \ldots, a_u\}\), where \(u > 1\). Let \(K_0\) and \(K\) be as follows

\[K_0: \quad v_1 := a_u;\]
\[K: \quad \text{if } (v_1 = a_0) \text{ then either } 0 \text{ or } v_1 := a_u; v_2 := a_u; \text{ or } v_1 := a_{u-1}; v_2 := a_u; \text{ else } v_1 := a_1; v_2 := a_u;\]

For an arbitrary natural number \(i > 0\), if \(\mu = [\mu_1, \mu_2, \ldots, \mu_u]\) is an input probability distribution then as a result of realization of program \(K; K^i\) we obtain the following output probability distribution

\[
\mu' = K_0 K^i_{< \mathfrak{S}, \rho>}(\mu) = [0, \ldots, 0, 1 - 0.5^{(i-1)}, 0, \ldots, 0, 0.5^i, 0, \ldots, 0, 0.5^i].
\]

**Proof.** Let us assume that \(< \mathfrak{S}, \rho >\) is a fixed probabilistic structure (for \(L^+ \rho\)) with a finite at least 2-element set \(A = \{a_1, a_2, \ldots, a_u\}\). Let’s consider an arbitrary program \(K_0; K^i\) \((i \in \mathbb{N}_+\)) The set of possible valuations of program variables contains \(u^2\) elements: \(w_1 = (a_1, a_1), w_2 = (a_1, a_2), \ldots, w_u = (a_1, a_u), w_{u+1} = (a_2, a_1), w_{u+2} = (a_2, a_2), \ldots, w_{2u} = (a_2, a_u), \ldots, w_{u^2-u+1} = (a_u, a_1), w_{u^2-u+2} = (a_u, a_2), \ldots, w_{u^2} = (a_u, a_u).\) We carry out computations for the input probability distribution \(\mu = [\mu_1, \mu_2, \ldots, \mu_u]\). The proof of the Lemma 3.1 will proceed by induction on the length of programs.

(A) The base of induction.

First we shall justify that the realization of the program \(K_0; K\) leads to the probability distribution

\[
\mu' = K_0 K^i_{< \mathfrak{S}, \rho>}(\mu) = [0, \ldots, 0, 0.5^i, 0, \ldots, 0, 0.5^i].
\]

We shall determine the necessary probability distributions (cf. the Appendix).

\[
[\chi_1 := a_1]_{< \mathfrak{S}, \rho>}(\mu) = [\mu_1 + \mu_{u+1} + \ldots + \mu_{u^2-u+1}, \mu_2 + \mu_{u+2} + \ldots + \mu_{u^2-u+2}, \ldots, \mu_{u+1} + \ldots + \mu_{u^2}, 0, \ldots, 0]
\]

\[
[\chi_1 := a_{u-1}]_{< \mathfrak{S}, \rho>}(\mu) = [0, \ldots, 0, \mu_1 + \mu_{u+1} + \ldots + \mu_{u^2-u+1}, \mu_2 + \mu_{u+2} + \ldots + \mu_{u^2-u+2}, \ldots, \mu_{u+1} + \ldots + \mu_{u^2}, 0, \ldots, 0]
\]

11
Let's denote the subprogram \( v_1 := a_u; v_2 := a_u \); by \( N_1 \).

\[
N_1 := a_u \]_{3, p}(\mu) = [v_2 := a_u]_{3, p}(v_1 := a_u]_{3, p}(\mu) = [0, \ldots , 0, (\mu_1 + \mu_{a+1} + \ldots + \mu_{u^2-a+1}) + (\mu_2 + \mu_{a+2} + \ldots + \mu_{u^2-a+2}) + \ldots + (\mu_a + u^2-u+1 \text{ times } \mu_a)] = [0, \ldots , 0, 1]
\]

By \( N_2 \) we denote the subprogram \( v_1 := a_{u-1}; v_2 := a_u \);.

\[
N_2 := a_u \]_{3, p}(\mu) = [v_2 := a_u]_{3, p}(v_1 := a_{u-1}]_{3, p}(\mu) = [0, \ldots , 0, (\mu_1 + \mu_{a+1} + \ldots + \mu_{u^2-a+1}) + (\mu_2 + \mu_{a+2} + \ldots + \mu_{u^2-a+2}) + \ldots + (\mu_a + u^2-u+1 \text{ times } \mu_a), 0, \ldots , 0] = [0, \ldots , 0, 0, \ldots , 0, 1, 0, \ldots , 0]
\]

The subprogram \( v_1 := a_1; v_2 := a_u \); we denote by \( N_3 \).

\[
N_3 := a_u \]_{3, p}(\mu) = [v_2 := a_u]_{3, p}(v_1 := a_1]_{3, p}(\mu) = [0, \ldots , 0, (\mu_1 + \mu_{a+1} + \ldots + \mu_{u^2-a+1}) + (\mu_2 + \mu_{a+2} + \ldots + \mu_{u^2-a+2}) + \ldots + (\mu_a + u^2-u \text{ times } \mu_a), 0, \ldots , 0] = [0, \ldots , 0, 0, \ldots , 0, 1, 0, \ldots , 0]
\]

Let's denote the subprogram \textbf{either} \( N_1 \) \textbf{or} \( N_2 \) \textbf{ro} by \( E \).

\[
E := 0.5 * (N_1 := a_u \]_{3, p}(\mu) + 0.5 * (N_2 := a_u \]_{3, p}(\mu) = [0, \ldots , 0, (\mu_1 + \mu_{a+1} + \ldots + \mu_{u^2-a+1}) + (\mu_2 + \mu_{a+2} + \ldots + \mu_{u^2-a+2}) + \ldots + (\mu_a + u^2-u-1 \text{ times } \mu_a), 0, \ldots , 0, 1, 0, \ldots , 0]
\]

12
The set of probabilistic algorithmic formulas of PrAL$^+$ valid in a finite structure is undecidable

$$= \left[ 0, \ldots, 0, 0.5 \ast (\mu_1 + \ldots + \mu_2), 0, \ldots, 0, 0.5 \ast (\mu_1 + \ldots + \mu_2) \right] =$$

$$= \left[ 0, \ldots, 0, 0.5, 0, \ldots, 0, 0.5 \right]$$

$$\begin{array}{l}
\text{[(v1 = a_\mu)?}_{<3,\rho>} (\mu) = [0, \ldots, 0, \mu_{a_\mu}, \mu_{a_\mu}, \mu_{a_\mu}, \ldots, \mu_{a_\mu}]}
\end{array}$$

$$\begin{array}{l}
\text{走到} (v1 = a_\mu)?_{<3,\rho>} (\mu) = [\mu_1, \mu_2, \ldots, \mu_{a_\mu}, 0, \ldots, 0]
\end{array}$$

$$\begin{array}{l}
K_{<3,\rho>} (\mu) = E_{<3,\rho>} \left( (v1 = a_\mu)?_{<3,\rho>} (\mu) \right) + N_{<3,\rho>} \left( [\neg (v1 = a_\mu)?_{<3,\rho>} (\mu)] \right) =
\end{array}$$

$$\begin{array}{l}
\left[ 0, \ldots, 0, 0.5 \ast (\mu_{a_\mu} + \mu_{a_\mu} + \ldots + \mu_{a_\mu}), 0, \ldots, 0, 0.5 \ast (\mu_{a_\mu} + \mu_{a_\mu} + \ldots + \mu_{a_\mu}) \right] =
\end{array}$$

$$\begin{array}{l}
\text{走到} K_{<3,\rho>} (K_{<3,\rho>} (\mu)) = K_{<3,\rho>} (v1 := a_\mu)?_{<3,\rho>} (\mu) =
\end{array}$$

Finally

$$\begin{array}{l}
(K_0 <_{3,\rho} > (v1 := a_\mu)?_{<3,\rho>} (\mu)) =
\end{array}$$

$$\begin{array}{l}
(K_0_{<3,\rho>} (\mu) = K_{<3,\rho>} (v1 := a_\mu)?_{<3,\rho>} (\mu) =
\end{array}$$

(B) The inductive step.

The inductive assumption. For a certain natural number $k$, if $\mu = [\mu_1, \mu_2, \ldots, \mu_2]$, is an input probability distribution then as a result of realization of the program $K_0; K^k$ we obtain the following output probability distribution
\[ K_0 K^k < \mathcal{S} > (\mu) = \]
\[ = \left[ 0, \ldots, 0, (1 - 0.5^{(k-1)}) \ast (\mu_1 + \mu_2 + \ldots + \mu_u), 0, \ldots, 0, 0.5^k \ast (\mu_1 + \mu_2 + \ldots + \mu_u), 0, \ldots, 0, 0.5^k \right] = \]
\[ = \left[ 0, \ldots, 0, (1 - 0.5^{(k-1)}), 0, \ldots, 0, 0.5^k, 0, \ldots, 0, 0.5^k \right] = \]

We shall apply the inductive assumption to show that if we take \( \mu = [\mu_1, \mu_2, \ldots, \mu_u] \) as the input probability distribution then after the execution of the program \( K_0 : K^{k+1} \) we obtain the following output probability distribution
\[ K_0 K_{<3,P}^k > (\mu) = \left[ 0, \ldots, 0, (1 - 0.5^k) \ast (\mu_1 + \mu_2 + \ldots + \mu_u), 0, \ldots, 0, 0.5^{(k+1)} \ast (\mu_1 + \mu_2 + \ldots + \mu_u) \right] = \]
\[ = \left[ 0, \ldots, 0, (1 - 0.5^k), 0, \ldots, 0, 0.5^{(k+1)}, 0, \ldots, 0, 0.5^{(k+1)} \right] = \]

We can express a composition of programs in the following way (cf. the Appendix)
\[ K_0 K_{<3,P}^k > (\mu) = K_{<3,P} (K_0 K_{<3,P}^k > (\mu)) \]

Hence by the inductive assumption
\[ K_{<3,P} (K_0 K_{<3,P}^k > (\mu)) = K_{<3,P} \left( 0, \ldots, 0, (1 - 0.5^{(k-1)}) \ast (\mu_1 + \mu_2 + \ldots + \mu_u), 0, \ldots, 0, 0.5^{(k-1)} \ast (\mu_1 + \mu_2 + \ldots + \mu_u) \right) = \]
\[ = \left[ 0, \ldots, 0, (1 - 0.5^{(k-1)}), 0, \ldots, 0, 0.5^{(k-1)} \ast (\mu_1 + \mu_2 + \ldots + \mu_u) \right] = \]
\[ = \left[ 0, \ldots, 0, (1 - 0.5^{(k-1)}), 0, \ldots, 0, 0.5^{(k-1)} \times 1 \times 1 \times \ldots \times 1 \ast (\mu_1 + \mu_2 + \ldots + \mu_u) \right] = \]

which accomplishes the inductive proof.

\[ \square \]

**Lemma 3.2** Let \( < \mathcal{S}, P > \) be an arbitrary fixed structure (for \( \mathcal{L}_A^{+} \)) with a finite set \( A = \{ a_1, a_2, \ldots, a_u \} \), where \( u > 1 \). The set of formulas of \( \text{PrAL}^+ \) valid in \( < \mathcal{S}, P > \) is undecidable.
The set of probabilistic algorithmic formulas of \( \text{PrAL}^+ \) valid in a finite structure is undecidable

**Proof.** Let \( < \mathfrak{A}, \rho > \) be an arbitrary fixed structure (for \( \text{L}^+_p \)) with a finite at least 2-element set \( A = \{ a_1, \ldots, a_u \} \). Let’s consider the formula \( \beta \) of the form \( K_0 \cup K\alpha \), where \( K_0, K \) are the programs considered in the Lemma 3.1 and \( \alpha \) is as follows

\[ \alpha: x = P(v_1 = a_{u-1} \land v_2 = a_u). \]

The computations are carried out for the input probability distribution \( \mu = [\mu_1, \mu_2, \ldots, \mu_2] \) and for programs \( K_0 \) and \( K_0; K^i \), where \( i \in N_+ \). Let’s denote\( K_0<3, \rho>(\mu) \) by \( \eta \). We know that

\[ \eta = K_0<3, \rho>(\mu) = [v_1 := a_u]<3, \rho>(\mu) = \left[ \begin{array}{c} 0, \ldots, 0, \mu_1 + \mu_{u+1} + \ldots + \mu_{a^2-u+1}, \mu_2 + \\ \quad \mu_{u+2} + \ldots + \mu_{a^2-u+2}, \ldots, \mu_u + \mu_{2u} + \ldots + \mu_{a^2} \end{array} \right]. \]

By the Lemma 3.1 we obtain that for an arbitrary number \( i > 0 \)

\[ \mu' = K_0K^i<3, \rho>(\mu) = \left[ \begin{array}{c} 0, \ldots, 0, (1 - 0.5^{(i-1)}) \\ \quad 0, \ldots, 0 \\ \quad 0.5^i, 0, \ldots, 0, 0.5^i \end{array} \right]. \]

We recall, that \( P(v_1 = a_{u-1} \land v_2 = a_u) = \mu'(w_{a^2-u}), \) where \( w_{a^2-u} = (a_{u-1}, a_u) \). We can notice that for \( i \in N_+ \) we have \( \mu'(w_{a^2-u}) = 0.5^i \) and additionally \( \eta(w_{a^2-u}) = 0 \). Therefore the formula \( \beta: K_0 \cup K\alpha \) describes the following fact

\[ (x = 0) \lor (x = 0.5) \lor (x = 0.25) \lor (x = 0.125) \lor \ldots \]

Let’s notice, that we can define an arbitrary natural number \( k \) in the following way. Let \( k \) be a real number

\[ N(k) \text{ iff } < \mathfrak{A}, \rho > \models (k = 0 \lor \exists x ((k = -1g x) \land K_0 \cup K\alpha)). \]

Since the natural numbers were generated among real numbers and operations of addition and multiplication exist in the structure \( \Re = < R; +, -, \ast, 0, 1, \rangle \), we can define these operations for constructed natural numbers. For arbitrary \( x_0, x_1, x_2 \)

\[ x_0 + x_1 = x_2 \text{ iff } < \mathfrak{A}, \rho > \models N(x_0) \land N(x_1) \land x_2 = x_0 + x_1; \]

\[ x_0 \ast x_1 = x_2 \text{ iff } < \mathfrak{A}, \rho > \models N(x_0) \land N(x_1) \land x_2 = x_0 \ast x_1. \]

Since \( Th(< N; +, \ast, 0, 1 >) \) is undecidable (cf. [2,11,7]), the set of formulas of considered algorithmic logic, valid in a fixed, finite at least 2-element structure (for \( \text{L}^+_p \)) is also undecidable. \( \square \)
4. Appendix (cf. [6])

By the interpretation of a program $K$ of $L_+^\rho$ in the structure $<\mathfrak{S}, \rho>$ we mean a function $K_{<\mathfrak{S}, \rho>} : \mathcal{M} \mapsto \mathcal{M}$ which is defined recursively.

- If $K$ is an assignment instruction of the form $v_r := \tau$ (for $v_r \in V$, $r = 1, \ldots, h$ and $\tau \in T$) then
  $[v_r := \tau]_{<\mathfrak{S}, \rho>} (\mu) = \mu'$, where
  $\mu'(w_j) = \sum_{w \in W} \mu(w)$ for $j = 1, \ldots, n$ and
  $\mathcal{W}^\tau = \{ w \in W : w(v_r) = \tau_3 (w_{in}) \wedge \forall v \in V \setminus \{v_r\} w(v) = w_{in}(v) \}.$
  $w_{in}$ denotes an input valuation of program variables.

- If $K$ is a random assignment of the form $v_r := \text{random}$ (for $v_r \in V$, $r = 1, \ldots, h$ and $\rho$ being a probability distribution defined on $A$) then
  $[v_r := \text{random}]_{<\mathfrak{S}, \rho>} (\mu) = \mu'$, where
  $\mu'(w_j) = \rho_i (w_j (v_r)) \cdot \sum_{w \in W} \mu(w)$ and
  $\mathcal{W}' = \{ w \in W : \forall v \in V \setminus \{v_r\} w(v) = w_{in}(v) \}.$

- We interpret the program while $\neg \gamma$ do $v := v$ od (for $v \in V$ and $\gamma \in F_\rho$) in the following way
  $[\gamma']_{<\mathfrak{S}, \rho>} (\mu) = [\text{while } \neg \gamma \text{ do } v := v \text{ od}]_{<\mathfrak{S}, \rho>} (\mu) = \mu'$, where
  $\mu'(w_j) = \begin{cases} \mu(w_i) & \text{for } w_i = w_j \wedge \mathfrak{S}, w_i \models \gamma \\ 0 & \text{otherwise} \end{cases}$
  We denote this program construction by $[\gamma']$.

- If $K$ is a composition of programs $M_1, M_2$ and $M_1_{<\mathfrak{S}, \rho>} (\mu), M_2_{<\mathfrak{S}, \rho>} (\mu)$ are defined then
  $[M_1; M_2]_{<\mathfrak{S}, \rho>} (\mu) = M_2_{<\mathfrak{S}, \rho>} (M_1_{<\mathfrak{S}, \rho>} (\mu)).$

- If $K$ is a branching between the two programs $M_1, M_2$ and $M_1_{<\mathfrak{S}, \rho>} (\mu), M_2_{<\mathfrak{S}, \rho>} (\mu)$ are defined then
  $[[\text{if } \gamma \text{ then } M_1 \text{ else } M_2 \text{ fi}]]_{<\mathfrak{S}, \rho>} (\mu) =$
  $= M_1_{<\mathfrak{S}, \rho>} ([\gamma']_{<\mathfrak{S}, \rho>} (\mu)) + M_2_{<\mathfrak{S}, \rho>} ([\neg \gamma']_{<\mathfrak{S}, \rho>} (\mu))$.

- If $K$ is a probabilistic branching, $p \in R$, $0 < p < 1$ and $M_1_{<\mathfrak{S}, \rho>} (\mu), M_2_{<\mathfrak{S}, \rho>} (\mu)$ are defined then
  $[\text{either}_p M_1 \text{ or } M_2 \text{ re}]]_{<\mathfrak{S}, \rho>} (\mu) = p \cdot M_1_{<\mathfrak{S}, \rho>} (\mu) + (1 - p) \cdot M_2_{<\mathfrak{S}, \rho>} (\mu).$
The set of probabilistic algorithmic formulas of PrAL⁺ valid in a finite structure is undecidable

References

Streszczenie Rozważamy probabilistyczną logikę algorytmiczną. W pracy [6] znajduje się uzasadnienie, że zbiór formuł logiki PrAL, prawdziwych w skończonej strukturze, jest rozstrzygalny ze względu na diagram struktury. Dodajemy do języka L⁺ logiki PrAL znak $\bigcup$ i funkтор lg. Następnie uzasadniamy, że zbiór formuł rozszerzonej logiki, prawdziwych w skończonej co najmniej 2-elementowej strukturze (dla L⁺), nie jest już rozstrzygalny.

Słowa kluczowe: probabilistyczna logika algorytmiczna, egzystencjalny kwantyfikator iteracji