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 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 \( \mathcal{I} \) (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 \( V_P = \{V \cup V_\mathbb{R} \bigcup V_0\} \) of variables, where a subset \( V = \{v_0, v_1, \ldots\} \) consists of non-arithmetic individual variables, a subset \( V_\mathbb{R} = \{x_0, x_1, \ldots\} \) contains real variables and a subset \( V_0 = \{q_0, q_1, \ldots\} \) contains propositional variables,
- a set of signs of relations \( \Psi_P = \{\Psi \cup \Psi_\mathbb{R}\} \), where the subset \( \Psi \) consists of non-arithmetic predicates and the subset \( \Psi_\mathbb{R} = \{<_\mathbb{R}, =_\mathbb{R}\} \) contains arithmetic predicates,
- an enumerable set of functors \( \Phi_P = \{\Phi \cup \Phi_\mathbb{R}\} \), which consists of the subset \( \Phi_\mathbb{R} = \{+, -, \times, \lg\} \) of symbols for arithmetic operations and the subset \( \Phi \) of symbols for non-arithmetic operations,
- the set \( \{\neg, \land, \lor, \Rightarrow, \Leftrightarrow\} \) of logical connectives,
- the set \( \{\text{if}, \text{then}, \text{else}, \text{fi}, \text{while}, \text{do}, \text{od}, \text{either}, \text{or}, \text{random} \}^1 \) 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 \(\tau_1, \ldots, \tau_n \in T\) then an expression \(\phi_i(\tau_1, \ldots, \tau_n)\) 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 \(t_1 + t_2, t_1 - t_2, t_1 \ast t_2, \lg t_1\) 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 \(\tau_1, \ldots, \tau_m \in T\) and \(\psi_j \in \Psi\) (\(\psi_j\) – an \(m_j\)-argument predicate) then \(\psi_j(\tau_1, \ldots, \tau_m) \in F_O\),

– 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 \(v := \text{random}\), 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 \(M_2\) \(\text{fi}\), 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 \(\neg \alpha, \alpha \lor \beta, \alpha \land \beta, \alpha \Rightarrow \beta, \alpha \Leftrightarrow \beta\) belong to \(F_P\).
– if \( \alpha \in F_p \) and \( x \in V_\mathbb{R} \) 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 \(< \{ n_k \}_{k \in \Phi}, \{ m_l \}_{\psi \in \Psi} >\) and let a relational system \( \mathcal{I} = < \mathcal{A} \cup R; \{ \phi_k \}_{k \in \Phi}, \{ \psi_l \}_{\psi \in \Psi} >\) (which consists of the fixed, finite, at least 2-element set \( \mathcal{A} \), the set \( R \) of real numbers, operations and relations) be a fixed data structure for \( L^+_p \).

We interpret non-arithmetic individual variables of \( L^+_p \) as elements of \( \mathcal{A} \). Real variables are interpreted as elements of the set \( R \) 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 \( \mathcal{I} \) we mean a function \( \tau_\mathcal{I} : W \rightarrow A \) which is defined recursively.

- If \( \tau \) is a variable \( v \in V \) then \( v_\mathcal{I}(w) = w(v) \).
- 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 \( \phi(\tau_1, \ldots, \tau_n)_\mathcal{I}(w) = \phi_\mathcal{I}(\tau_1_\mathcal{I}(w), \ldots, \tau_n_\mathcal{I}(w)) \), where \( \tau_1_\mathcal{I}(w), \ldots, \tau_n_\mathcal{I}(w) \) 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 \( A \)

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

**Definition 2.7** (cf. [6]) A pair \(< \mathcal{I}, \rho >\), where \( \rho \) is a set of fixed probability distributions \( \rho_i \) defined on \( A \) and \( \mathcal{I} \) 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

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

By \( S \) we mean the set of all states, i.e. all pairs \( s = < \mu, w_\mathbb{R} >\), where \( \mu \) is a probability distribution of valuations of non-arithmetic variables and \( w_\mathbb{R} \) is a valuation of real variables of \( V_\mathbb{R} \).
The set of probabilistic algorithmic formulas of PrAL+ valid in a finite structure is undecidable

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

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

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])

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

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])

$$[P(\alpha)]_{<\mathcal{I}, \rho>} (s) = \sum_{w \in W^\alpha} \mu(w),$$ 

where $W^\alpha = \{w \in \mathcal{I} : \mathcal{I}, w \models \alpha\}$.

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

Given below is the interpretation of a formula $K\alpha$ ($\alpha \in F_p$ and $K \in \Pi$).

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

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

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

The next definition establishes the meaning of the existential iteration quantifier ($K \in \Pi, \alpha \in F_p$).
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 $\bigcup K\alpha$ ($K \in \Pi$, $\alpha \in F_{\Pi}$) 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 $\beta$:

$$K_0 \bigcup K\alpha \text{ such that }$$

$$K_0 : \; v_1 := 0;$$

$$K : \; \text{if } (v_1 = 0) \text{ then } v_1 := \text{random}; \; v_2 := 0; \text{ else } v_2 := 1; \text{ fi}$$

$$\alpha : \; x = P(v_1 = 1 \lor v_2 = 0)$$

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 $v_1 := \text{random}$ 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 $\lg$ (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 $<\mathcal{S},\rho>$ be an arbitrary fixed probabilistic structure (for $L_0^+$) with a finite set $A = \{a_1, a_2, \ldots, a_u\}$, where $u > 1$. Let $K_0$ and $K$ be as follows

$K_0$: $v_1 := a_u$;
$K$: if $(v_1 = a_u)$ then either $v_1 := a_u, v_2 := a_u$; or $v_1 := a_{u-1}, v_2 := a_{u-1}$.

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 $K_0; K'$ we obtain the following output probability distribution

$$\mu' = K_0 K_{<\mathcal{S},\rho>}(\mu) = \left[ 0, \ldots, 0, 1 - 0.5^{(i-1)}, 0, \ldots, 0, 0.5, 0, \ldots, 0, 0.5 \right].$$

Proof. Let us assume that $<\mathcal{S},\rho>$ is a fixed probabilistic structure (for $L_0^+$) with a finite at least 2-element set $A = \{a_1, a_2, \ldots, a_u\}$. Let’s consider an arbitrary program $K_0; K'$ ($i \in \mathbb{N}_+$). The set of possible valuations of program variables contains $u^2$ elements: $w_1 = (a_1, a_1)$, $w_2 = (a_1, a_2)$, ..., $w_u = (a_1, a_u)$, $w_{u+1} = (a_2, a_1)$, $w_{u+2} = (a_2, a_2)$, ..., $w_{2u} = (a_2, a_u)$, ..., $w_{u^2-u+1} = (a_u, a_1)$, $w_{u^2-u+2} = (a_u, a_2)$, ..., $w_{u^2} = (a_u, a_u)$. 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

$$\mu' = K_0 K_{<\mathcal{S},\rho>}(\mu) = \left[ 0, \ldots, 0, 0.5, 0, \ldots, 0, 0.5 \right].$$

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

$$[v_1 := a_1]_{<\mathcal{S},\rho>}(\mu) = [\mu_1 + \mu_{u+1} + \ldots + \mu_{u^2-u+1}, \mu_2 + \mu_{u+2} + \ldots + \mu_{u^2-u+2}, \ldots, \mu_u + \mu_{2u} + \ldots + \mu_{u^2}, 0, \ldots, 0]$$

$$[v_1 := a_{u-1}]_{<\mathcal{S},\rho>}(\mu) = [0, \ldots, 0, \mu_1 + \mu_{u+1} + \ldots + \mu_{u^2-u+1}, \mu_2 + \mu_{u+2} + \ldots + \mu_{u^2-u+2}, \ldots, \mu_u + \mu_{2u} + \ldots + \mu_{u^2}, 0, \ldots, 0]$$

$$[v_1 := a_{u-1}]_{<\mathcal{S},\rho>}(\mu) = [0, \ldots, 0, \mu_1 + \mu_{u+1} + \ldots + \mu_{u^2-u+1}, \mu_2 + \mu_{u+2} + \ldots + \mu_{u^2-u+2}, \ldots, \mu_u + \mu_{2u} + \ldots + \mu_{u^2}, 0, \ldots, 0]$$

$$[v_1 := a_{u-1}]_{<\mathcal{S},\rho>}(\mu) = [0, \ldots, 0, \mu_1 + \mu_{u+1} + \ldots + \mu_{u^2-u+1}, \mu_2 + \mu_{u+2} + \ldots + \mu_{u^2-u+2}, \ldots, \mu_u + \mu_{2u} + \ldots + \mu_{u^2}, 0, \ldots, 0]$$

11
\[ v_1 := a_d \mu(\mu) = \left[ 0, \ldots, 0, \mu_1 + \mu_u + \ldots + \mu_{u^3-u+1}, \mu_2 + \mu_u + \ldots + \mu_{u^2-u+2}, \ldots + \right. \]
\[ \left. a^2-u \text{ times} \right] \]
\[ v_2 := a_d \mu(\mu) = \left[ 0, \ldots, 0, \mu_1 + \mu_2 + \ldots + \mu_u, 0, \ldots, 0, \mu_u + \mu_2 + \ldots + \mu_{u^2-u+2}, \ldots + \mu_{u^2-u+2}, \ldots + \mu_{u^2-u+2}, \ldots + \right. \]
\[ \left. u-1 \text{ times} \mu_{u^2-u+2} \text{ times} \right] \]

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

\[ N_{1,3,\mu}(\mu) = [v_2 := a_d] \mu(\mu) \]
\[ = [0, \ldots, 0, (\mu_1 + \mu_u + \ldots + \mu_{u^3-u+1}) + (\mu_2 + \mu_u + \ldots + \mu_{u^2-u+2}) + \ldots + (\mu_u + \mu_{u^2-u+2} + \ldots + \mu_{u^2-u+2}) + \ldots + (\mu_u + \mu_{u^2-u+2} + \ldots + \mu_{u^2-u+2})] = [0, \ldots, 0, 1] \]
\[ u-1 \text{ times} \mu_{u^2-u+2} \text{ times} \]

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

\[ N_{2,3,\mu}(\mu) = [v_2 := a_d] \mu(\mu) \]
\[ = [0, \ldots, 0, (\mu_1 + \mu_u + \ldots + \mu_{u^3-u+1}) + (\mu_2 + \mu_u + \ldots + \mu_{u^2-u+2}) + \ldots + (\mu_u + \mu_{u^2-u+2} + \ldots + \mu_{u^2-u+2}) + \ldots + (\mu_u + \mu_{u^2-u+2} + \ldots + \mu_{u^2-u+2})] = [0, \ldots, 0, 1, 0, \ldots, 0] \]
\[ u-1 \text{ times} \mu_{u^2-u+2} \text{ times} \]

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

\[ N_{3,3,\mu}(\mu) = [v_2 := a_d] \mu(\mu) \]
\[ = [0, \ldots, 0, (\mu_1 + \mu_u + \ldots + \mu_{u^3-u+1}) + (\mu_2 + \mu_u + \ldots + \mu_{u^2-u+2}) + \ldots + (\mu_u + \mu_{u^2-u+2} + \ldots + \mu_{u^2-u+2}) + \ldots + (\mu_u + \mu_{u^2-u+2} + \ldots + \mu_{u^2-u+2})] = [0, \ldots, 0, 1, 0, \ldots, 0] \]
\[ u-1 \text{ times} \mu_{u^2-u+2} \text{ times} \]

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

\[ E_{3,\mu}(\mu) = 0.5 * (N_{1,3,\mu}(\mu)) + 0.5 * (N_{2,3,\mu}(\mu)) = \]
\[ = 0.5 * \left[ 0, \ldots, 0, \mu_1 + \mu_2 + \ldots + \mu_{u^2}, 0, 0, \ldots, 0 \right] + 0.5 * \left[ 0, \ldots, 0, \mu_1 + \mu_2 + \ldots + \mu_{u^2}, 0, 0, \ldots, 0 \right] = \]
\[ u-1 \text{ times} \mu_{u^2-u} \text{ times} u \times \]
The set of probabilistic algorithmic formulas of PrAL⁺ valid in a finite structure is undecidable

\[
K_{<3,p>}(\mu) = E_{<3,p>}([\mu]_{<3,p>}(\mu)) + N_{3<3,p>}(\mu) = \mu_{u} + \mu_{v}.
\]

Finally

\[
K_{<3,p>}(K_{<3,p>}(\mu)) = K_{<3,p>}(\mu) = [0, \ldots, 0, 0.5 \times (\mu_1 + \ldots + \mu_{u})].
\]

(B) The inductive step.

The inductive assumption. For a certain natural number \(k\), if \(\mu = [\mu_1, \mu_2, \ldots, \mu_v]\) 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
$K_0 K^k < \mathcal{S}, \rho > (\mu) =$

\[
= \left[0, \ldots, 0, (1 - 0.5^{(k-1)}) \cdot (\mu_1 + \mu_2 + \ldots + \mu_{2^u}), 0, \ldots, 0, 0.5^k \cdot (\mu_1 + \mu_2 + \ldots + \mu_{2^u}) \right]
\]

\[
= \left[0, \ldots, 0, 0, 0.5^k \cdot (\mu_1 + \mu_2 + \ldots + \mu_{2^u}) \right]
\]

We shall apply the inductive assumption to show that if we take $\mu = [\mu_1, \mu_2, \ldots, \mu_{2^u}]$ as the input probability distribution then after the execution of the program $K_0 ; K^{k+1}$ we obtain the following output probability distribution

\[
K_0 K^{k+1} < \mathcal{S}, \rho > (\mu) = \left[0, \ldots, 0, (1 - 0.5^k) \cdot (\mu_1 + \mu_2 + \ldots + \mu_{2^u}), 0, \ldots, 0, 0.5^k \cdot (\mu_1 + \mu_2 + \ldots + \mu_{2^u}) \right]
\]

\[
= \left[0, \ldots, 0, 0.5^k \cdot (\mu_1 + \mu_2 + \ldots + \mu_{2^u}) \right]
\]

We can express a composition of programs in the following way (cf. the Appendix)

$K_0 K^{k+1} < \mathcal{S}, \rho > (\mu) = K_0 K^k < \mathcal{S}, \rho > (K_0 K^{k+1} < \mathcal{S}, \rho > (\mu))$

Hence by the inductive assumption

\[
K_0 K^{k+1} < \mathcal{S}, \rho > (\mu) = K_0 K^k < \mathcal{S}, \rho > \left( \left[0, \ldots, 0, (1 - 0.5^{(k-1)}) \cdot (\mu_1 + \mu_2 + \ldots + \mu_{2^u}), 0, \ldots, 0, 0.5^k \cdot (\mu_1 + \mu_2 + \ldots + \mu_{2^u}) \right] \right)
\]

\[
= \left[0, \ldots, 0, (1 - 0.5^{(k-1)}) + 0.5^k, 0, \ldots, 0, 0.5 \cdot 0.5^k \cdot 0, \ldots, 0, 0.5^k \cdot 0.5^k \right]
\]

\[
= \left[0, \ldots, 0, (1 - 0.5^{(k-1)}) + 0.5^k, 0, \ldots, 0, 0.5 \cdot 0.5^k \cdot 0, \ldots, 0, 0.5^k \cdot 0.5^k \right]
\]

which accomplishes the inductive proof.

\[\square\]

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

**Proof.** Let \( < \mathfrak{S}, \rho > \) be an arbitrary fixed structure (for \( L_\rho^+ \)) with a finite at least 2-element set \( A = \{ a_1, \ldots, a_n \} \). 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 \( \mu = [\mu_1, \mu_2, \ldots, \mu_\mu^2] \) and for programs \( K_0 \) and \( K; K' \), where \( i \in N_+ \). Let’s denote \( K_{0<3, \rho>} \) by \( \eta \). We know that

\[ \eta = K_{0<3, \rho>} (\mu) = [v_1 := a_u]_{<3, \rho>} (\mu) = \begin{bmatrix} 0, \ldots, 0, & \mu_1 + \mu_{u+1} + \ldots + \mu_{u^2-u+1}, & \mu_2 + \\
\mu_{u+2} + \ldots + \mu_{u^2-u+2}, & \ldots, & \mu_u + \mu_{2u} + \ldots + \mu_{\mu^2} \end{bmatrix}. \]

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

\[ \mu' = K_{0,3, \rho>} (\mu) = \begin{bmatrix} 0, \ldots, 0, & (1 - 0.5^{(i-1)}) \times \mu, \ldots, 0, & 0.5^i, \ldots, 0, 0.5^i \end{bmatrix}. \]

We recall, that \( P(v_1 = a_{u-1} \land v_2 = a_u) = \mu' (w_{u^2-u}), \) where \( w_{u^2-u} = (a_{u-1}, a_u) \). We can notice that for \( i \in N_+ \) we have \( \mu' (w_{u^2-u}) = 0.5^i \) and additionally \( \eta(w_{u^2-u}) = 0 \).

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

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

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

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

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

Since \( Th(< N; +, *, 0, 1 >) \) 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 \( L_\rho^+ \)) is also undecidable.

\[ \square \]
4. Appendix (cf. [6])

By the interpretation of a program $K$ of $L_p^+$ in the structure $<\mathcal{S}, \rho>$ we mean a function $K_{\langle \mathcal{S}, \rho \rangle} : \mathcal{M} \mapsto \mathcal{M}$ 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
  \[
  [v_r := \tau]_{\langle \mathcal{S}, \rho \rangle}(\mu) = \mu', \text{ where}
  \]
  \[
  \mu'(w_j) = \sum_{w \in W^j} \mu(w) \text{ for } j = 1, \ldots, n \text{ and}
  \]
  \[
  W^j = \{ w \in W : w(v_r) = \tau_{\mathcal{S}}(w_{in}) \land \forall v \in V \setminus \{v_r\} w(v) = w_{in}(v) \}. \]
  \[
  w_{in} \text{ denotes an input valuation of program variables.}
  \]

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

- We interpret the program while $\neg \gamma$ do $v := v$ od (for $v \in V$ and $\gamma \in F_0$) in the following way
  \[
  [\gamma]_{\langle \mathcal{S}, \rho \rangle}(\mu) = \text{while } \neg \gamma \text{ do } v := v \text{ od}_{\langle \mathcal{S}, \rho \rangle}(\mu) = \mu', \text{ where}
  \]
  \[
  \mu'(w_j) = \begin{cases} \mu(w_j) & \text{for } w_j = w_j(\mathcal{S}, w_i) = \gamma \\ 0 & \text{otherwise} \end{cases}, \]
  \[
  \text{We denote this program construction by } [\gamma \cdot]. \]

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

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

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

References

ZBIÓR FORMUŁ LOGIKI PrAL⁺ PRAWDZIWYCH W SKOŃCZONEJ STRUKTURZE JEST NIEROZSTRZYGALNY

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 $L_{P}$ logiki PrAL znak $\bigcup$ i funktor $lg$. Następnie uzasadniamy, że zbiór formuł rozszerzonej logiki, prawdziwych w skończonej co najmniej 2-elementowej strukturze (dla $L_{P}^{+}$), nie jest już rozstrzygalny.

Słowa kluczowe: probabilistyczna logika algorytmiczna, egzystencjonalny kwantyfikator iteracji