ДОКЛАДЫ АКАДЕМИИ НАУК, *2019*, *том 485*, *№ 1*, *с. 11–14*

МАТЕМАТИКА

УДК 510.2; 510.3; 510.6; 51-8; 004.8

ЛОГИЧЕСКИЙ ЯЗЫК ОПИСАНИЯ ПОЛИНОМИАЛЬНОЙ ВЫЧИСЛИМОСТИ

Академик РАН С. С. Гончаров*, Д. И. Свириденко**

Поступило 16.10.2018 г.

Построено расширение понятия терма и соответственно формулы за счёт новых конструкций. Построены расширения языка, сохраняющие выразительность на Σ-формулах, а на уровне Δ_0-формул и термов, обеспечивающих полиномиальность алгоритмов вычислений значений терма и истинности Δ_0-формулы.

Ключевые слова: логика, ограниченные кванторы, наследственно конечные списки, вычислимая модель, терм, условные термы, рекурсивные термы, формула, Σ-формула, Δ_0-формула, полиномиальные алгоритмы, полиномиальная модель, семантическое программирование.

DOI: https://doi.org/10.31857/S0869-5652485111-14

Исследования по построению языка семантического программирования на основе дескриптивных конструкций для вычислимых функций и отношений были начаты в работах [1–5], где предложена конструкция построения языка программирования логического типа. Началу этих исследований послужили работы Ю.Л. Ершова [6–8], посвящённые построению обобщённой теории вычислимости на основе теории допустимых множеств Барвайса [9] над расширением исходной модели данных надстройкой из наследственно конечных множеств. При рассмотрении вопросов применения данного подхода к построению систем программирования была построена теория вычислимости через Σ-определимость в надстройке из наследственно конечных списков [2]. При исследовании вопросов полиномиальной сложности будем использовать методы и понятия из [10, 11].

В работе [12] было доказано, что проверка истинности для каждой Δ_0-формулы, а также вычисление значений терма на супермодели наследственно конечных списков над моделью полиномиальной сложности P-, PSPACE-сложности также будет P-, PSPACE-полиномиальной. В этой связи возникла идея построения логического языка, в котором это свойство сохранялось бы, но Δ_0-термы, построенные с использованием Δ_0-формул, определяли бы все полиномиально вычислимые функции, сохраняя при этом декларативную ясность их определений. В данной работе рассматриваются вопросы определимости на основе Δ_0-формул, для которых вопросы проверки истинности имеют ограниченную сложность, как применительно к базисным термам и отношениям в основной модели, так и относительно реализации списочных операций в суперструктуре. С точки зрения конкретных приложений этой логической системы программирования можно различить два типа решаемых задач. К первому типу относятся локальные задачи построения конкретных вычислений с данными исследуемой области, требующие наличия быстрых способов вычисления этих характеристик по принятию оперативных решений в режиме онлайн. Второй тип задач — это поиск стратегических многоцелевых решений для умных контрактов (smart contract) и принятия решений, которые используют при своём выборе большие данные и требуют поиска и задания уже на языке, допускающем неограниченные кванторы существования.

Для решения задач первого типа в работе [13] было предложено расширить класс термов нашего Σ-языка условными термами, которые могут определяться с применением Δ_0-формул. В работе было замечено, что такое обогащение понятий Δ_0-терма и Δ_0-формулы сохраняет полиномиальность и расширяет выразительные возможности термов для возникающих при обработке данных задач, требующих применимости оператора If-then-else.

В настоящей работе предпринято исследование новых дополнительных конструкций, введение которых сохраняет как полиномиальную сложность термальных функций, так и полиномиальность алгоритма проверки истинности для фиксированных Δ_0-формул. Оказалось, что набор новых конструкций при этом является достаточным для возникающих проблем в интеллектуальных системах при их спе-

Институт математики им. С.Л. Соболева Сибирского отделения Российской Академии наук, Новосибирск

**E-mail*: *s.s.goncharov@math.nsc.ru*

***E-mail*: *dsviridenko47@gmail.com*

цификации в рамках построенного логического формализма.

Рассмотрим модель $\mathfrak{M}$ сигнатуры σ и определим, следуя [1] и [3], надстройку наследственно конечных списков $HW(\mathfrak{M})$ для модели $\mathfrak{M}$, расширив сигнатуру σ до σ^*, добавляя стандартные списочные функции языка LISP: head(x), cons(x, y), tail(x, y) и константу nil, а также двуместные предикаты: $\in$ — "быть элементом списка" и $\sqsubseteq$ — "быть начальным сегментом списка", одноместный предикат $\mathcal{U}$, выделяющий основное множество базисной структуры $\mathfrak{M}$. Отношения $\in$ и $\sqsubseteq$ определяют частичные отношения порядка $\leq_\in$ и $\leq_\sqsubseteq = \sqsubseteq$.

Если мы желаем работать в нашем языке с матричными данными, то нам потребуются в задании языковых конструкций, требуемых, скажем, при машинном обучении, уже рекурсивные процедуры. Для этих целей было рассмотрено обогащение конструкции термов рекурсией по спискам. Однако при таком обогащении может увеличиться и класс термальных функций. Так, например, если при примитивно рекурсивном представлении базисной модели класс термальных функций не увеличивается, то при полиномиальном P-, PSPACE-представлении класс термальных функций при использовании рекурсивной схемы уже расширяется до класса примитивно рекурсивных.

В настоящей работе предлагаются два варианта обогащения для построения термов, которые при полиномиальном представлении базисной модели не расширяют класс термальных функций полиномиальной P-, PSPACE-сложности.

Для обогащения $HW(\mathfrak{M})$ модели $\mathfrak{M}$ рассмотрим класс термов с переменными из V и классы Δ_0-формул, Δ-формул и Σ-формул [1]. Расширим понятие терма в сигнатуре σ^* до класса $\Delta_0(B\text{-list})(\mathfrak{B})$-термов, а класс Δ_0-формул — до класса $\Delta_0(B\text{-list})(\mathfrak{B})$-формул, которые определим по индукции, и эти обогащения будут привязаны к нашей модели $HW(\mathfrak{M})$.

Базис индукции: термы сигнатуры σ^* списочной надстройки являются $\Delta_0(\mathfrak{B})$-термами, а Δ_0-формулы являются $\Delta_0(B\text{-list})(\mathfrak{B})$-формулами.

Шаг индукции:

1. Если t, q — $\Delta_0(B\text{-list})(\mathfrak{B})$-термы, то $(f = q)$ — $\Delta_0(B\text{-list})(\mathfrak{B})$-формула.

2. Если $t_1, \ldots, t_n$ — $\Delta_0(B\text{-list})(\mathfrak{B})$-термы, а P — n-местный предикатный символ сигнатуры σ, то $P(t_1, \ldots, t_n)$ — $\Delta_0(B\text{-list})(\mathfrak{B})$-формула.

3. Если Φ и Ψ — $\Delta_0(B\text{-list})(\mathfrak{B})$-формулы, то $\neg\Phi$, $(\Phi \vee \Psi)$, $(\Phi \mathbin{\&} \Psi)$, $(\Phi \to \Psi)$ — также $\Delta_0(B\text{-list})(\mathfrak{B})$-формулы.

4. Если x — переменная, t — $\Delta_0(B\text{-list})(\mathfrak{B})$-терм, а Φ — $\Delta_0(B\text{-list})(\mathfrak{B})$-формула, то $\exists x \leq_i t\Phi$, $\forall x \leq_i t\Phi$ — также $\Delta_0(B\text{-list})(\mathfrak{B})$-формулы, где $\leq_i \in \{\leq_\varepsilon, \sqsubseteq\}$.

5. Условный оператор построения термов. Если $t_0, t_1, \ldots, t_{n+1}$ — $\Delta_0(B\text{-list})(\mathfrak{B})$-термы, а $\varphi_0, \varphi_1, \ldots, \varphi_n$ — $\Delta_0(B\text{-list})(\mathfrak{B})$-формулы, то выражение $t(\mathbf{v}) \rightleftharpoons$ $\rightleftharpoons \mathrm{Cond}((t_0, \ldots, t_{n+1}), (\varphi_0, \ldots, \varphi_n))$ с семантикой

$$t(\mathbf{v}) = \begin{cases} t_0(\mathbf{v}), & \text{если } \mathfrak{M} \models \varphi_0(\mathbf{v}),\\ t_1(\mathbf{v}), & \text{если } \mathfrak{M} \models \varphi_1(\mathbf{v}) \mathbin{\&} \neg\varphi_0(\mathbf{v}),\\ \ldots & \\ t_i(\mathbf{v}), & \text{если } \mathfrak{M} \models \varphi_i(\mathbf{v}) \mathbin{\&} \neg\varphi_0(\mathbf{v}) \mathbin{\&} \ldots \mathbin{\&} \neg\varphi_{i-1}(\mathbf{v}),\\ \ldots & \\ t_n(\mathbf{v}), & \text{если } \mathfrak{M} \models \varphi_n(\mathbf{v}) \mathbin{\&} \neg\varphi_0(\mathbf{v}) \mathbin{\&} \ldots \mathbin{\&} \neg\varphi_{n-1}(\mathbf{v}),\\ t_{n+1}(\mathbf{v}), & \text{в противном случае} \end{cases}$$

является $\Delta_0(B\text{-list})(\mathfrak{B})$-термом.

Определим ещё несколько конструкций для образования термов с помощью ограниченной рекурсии.

6. Оператор ограниченной рекурсии по $\Delta_0(B\text{-list})(\mathfrak{B})$-терму t.

Если $f(x_1, \ldots, x_n)$, $h(x_1, \ldots, x_n, y, z)$ — $\Delta_0(B\text{-list})(\mathfrak{B})$-термы, а t — $\Delta_0(B\text{-list})(\mathfrak{B})$-терм без свободных вхождений переменных, то определим функцию g следующим образом:

а) $g(x_1, \ldots, x_n, a) = a$ для любого элемента a из базисной модели $\mathfrak{M}$;

б) $g(x_1, \ldots, x_n, \mathrm{nil}) = f(x_1, \ldots, x_n)$;

в) $g(x_1, \ldots, x_n, \mathrm{cons}(\alpha, b)) = h(x_1, \ldots, x_n, \alpha)b$ для списка cons(α, b).

Заметим, что так определённая функция $g(x_1, \ldots, x_n, t)$ на нашей супермодели наследственно конечных списков является всюду определённой. Запись Rec(t, f, h) с семантикой, определённой выше, объявляется $\Delta_0(B\text{-list})(\mathfrak{B})$-термом.

7. Оператор b-while выбора элемента из списка, заданного $\Delta_0(B\text{-list})(\mathfrak{B})$-термом t с заданным условием относительно отношений $\in$ или $\sqsubseteq$.

Если $\varphi(x_1, \ldots, x_n, x_{n+1})$ — $\Delta_0(B\text{-list})(\mathfrak{B})$-формула и t — $\Delta_0(B\text{-list})(\mathfrak{B})$-терм, то определим в качестве значения функции $G(x_1, \ldots, x_n)$ первый элемент $y \in t(x_1, \ldots, x_n)$ (первый начальный сегмент $y \sqsubseteq t(x_1, \ldots, x_n)$) такой, что выполнено для него условие $\varphi(x_1, \ldots, x_n, y)$, если же такого элемента нет, то полагаем $G(x_1, \ldots, x_n) = t(x_1, \ldots, x_n)$. Терм b-while (φ, t) с данной семантикой объявляется $\Delta_0(B\text{-list})(\mathfrak{B})$-термом.

8. Оператор $\Delta_0(B\text{-list})(\mathfrak{B})$-определимой функции с ограничением $\Delta_0(B\text{-list})(\mathfrak{B})$-термом t, предложенный А. Нетесовым.

Если $\Delta_0(B\text{-list})(\mathfrak{B})$-формула определяет в нашей модели график всюду определённой функции f и $\Delta_0(B\text{-list})(\mathfrak{B})$-терм t мажорирует нашу функцию относительно одного из отношений $\in$, $\sqsubseteq$, т.е. для любых значений $f(x) \leq_i t(x)$, где $i \in \{\in, \sqsubseteq\}$, то терм $\mathrm{Lim}(\varphi, t)$, представляющий функцию f, объявляется $\Delta_0(B\text{-list})(\mathfrak{B})$-термом.

Других $\Delta_0(B\text{-list})(\mathfrak{B})$-термов и $\Delta_0(B\text{-list})(\mathfrak{B})$-формул нет.

Из определённых таким образом $\Delta_0(B\text{-list})(\mathfrak{B})$-термов и $\Delta_0(B\text{-list})(\mathfrak{B})$-формул стандартным образом определим $\Sigma_0(B\text{-list})(\mathfrak{B})$-формулы.

Заметим, что из определения $\Delta_0(B\text{-list})(\mathfrak{B})$-термов и $\Delta_0(B\text{-list})(\mathfrak{B})$-формул можно исключить конструкцию 8, не уменьшая выразительные возможности этого языка в силу определимости конструкции Lim через конструкцию b-while.

Теорема 1. *Любая функция $HW(\mathfrak{B})$, определённая $\Delta_0(B\text{-list})$-термом t в $HW(\mathfrak{B})$, может быть определима с использованием b-while-оператора, но без использования конструкции 8, т.е. конструкции $\Delta_0(B\text{-list})$-определимой функции с ограничением $\Delta_0(B\text{-list})$-термом t относительно отношений $\in$ или $\sqsubseteq$.*

Заметим, что при определении расширения Δ_0-формул (Σ-формул) до $\Delta_0(B\text{-list})(\mathfrak{B})$-формул мы сохраняем полиномиальную вычислимость над полиномиальной моделью. С другой стороны, не любая функция с $\Delta_0(B\text{-list})(\mathfrak{B})$-определимым графиком будет вычислима за полиномиальное время. Для таких функций необходимо рассматривать конструкции, ограничивающие поиск.

Теорема 2. 1) *Для любого полиномиального* P-, PSPACE- *представления модели $HW(\mathfrak{B})$ и для любой $\Delta_0(\mathfrak{B})$-формулы φ существует полиномиальный* P-, PSPACE-*алгоритм, определяющий по формуле φ и любому набору* $\mathbf{v}$ *из модели $HW(\mathfrak{B})$ выполнимость* $HW(\mathfrak{M}) \vDash \varphi(\mathbf{v})$.

2) *Для любого полиномиального* P-, PSPACE-*представления модели $HW(\mathfrak{B})$ и для любого $\Delta_0(B\text{-list})(\mathfrak{B})$-терма $t(\mathbf{v})$ определённая этим термом функция на модели $HW(\mathfrak{B})$ будет полиномиальной* P-, PSPACE-*сложности*.

Заметим, что в любой модели $HW(\mathfrak{B})$ всегда есть множество натуральных чисел Nat, которое состоит из пустого списка и списков, состоящих из пустых списков. Прибавление единицы в Nat есть термальная операция, а отношение порядка определяется отношением “быть начальным списком”. Так, определённое множество Nat будет Δ_0-определимым. Добавим также в качестве базисных операций в модель $HW(\mathfrak{B})$ операции сложения и умножения на списочных представлениях натуральных чисел Nat. Используя эти функции, определим на модели $HW(\mathfrak{B})$ новые функции: $\mathrm{lh}(\lambda)$-функция, определяющая длину последовательности λ; $\mathrm{change}(\lambda, i, \beta)$-функция, заменяющая i-й элемент в последовательности λ на элемент β для $0 < i \leq lh(\lambda)$, а в других случаях выдающая λ без изменений; $\mathrm{conc}(\alpha, \beta)$-функция, определяющая конкатенацию списков λ и β, а также $\mathrm{cons}_1(\alpha, \beta)$, которая добавляет в список α элемент β в качестве первого элемента.

Теорема 3. *Для любой полиномиальной* P-, PSPACE- *модели $\mathfrak{M}$ существует полиномиальное* P-, PSPACE-*представление модели $HW(\mathfrak{B})$ с полиномиально* P-, PSPACE-*вычислимыми функциями*: $\mathrm{lh}(\lambda)$, $r(\lambda)$, $\mathrm{change}(\lambda, i, \beta)$, $\mathrm{conc}(\alpha, \beta)$, $\mathrm{cons}_1(\alpha, \beta)$.

Расширим нашу конструкцию построения термов ещё одним оператором “Iteration”. В этом случае мы расширим множества термов и формул до множеств $\Delta_0(\mathfrak{J})$-термов, $\Delta_0(\mathfrak{J})$-формул и $\Sigma(\mathfrak{J})$-формул, которые определяются индуктивно с добавлением новой конструкции.

9. Оператор “Iteration”: Пусть $g(\mathbf{x}, \upsilon)$ — $\Delta_0^*(\mathfrak{J})$-терм, и существуют $\Delta_0(\mathfrak{J})$-термы q и p, $\Delta_0(\mathfrak{J})$-формула ψ такие, что $r(g^i(\mathbf{x}, \upsilon))$ меньше $r(q(\mathbf{x}, \upsilon))$, и для любых $\mathbf{x}$, υ найдётся $i < (r(p(\mathbf{x}, \upsilon)))$ такое, что выполнено $\varphi(g^i(\mathbf{x}, \upsilon))$, где

$$g^0(\mathbf{x}, \upsilon) = g(\mathbf{x}, \upsilon)$$

и

$$g^{i+1}(\mathbf{x}, \upsilon) = g(\mathbf{x}, g^i(\mathbf{x}, \upsilon)).$$

В этом случае определим в качестве $\Delta_0(\mathfrak{J})$-терма $t(\mathbf{x}, \upsilon)$-конструкцию, которая принимает значение, равное $g^i(\mathbf{x}, \upsilon)$ для первого i такого, что выполнено $\varphi(g^i(\mathbf{x}, \upsilon))$. Заметим, что функция сложности r на списках, замкнутых термах и замкнутых формулах задаётся с целью оценки сложности строения данных выражений. Если задана некая полиномиальная нумерация υ модели $\mathfrak{M}$, то строим полиномиальное представление модели $HW(\mathfrak{B})$ и множества термов и Δ_0-формул, следуя конструкции из [12], а также полиномиальную функцию r на номерах списков, замкнутых термов и замкнутых формул, которая будет определять сложность этих конструкций.

Теорема 4. *Существует полиномиальный* P-, PSPACE-*алгоритм, строящий по любой $\Delta_0(\mathfrak{J})$-формуле φ такие Σ-формулы ψ_1 и ψ_2, что*

$$HW(\mathfrak{M}) \models (\forall \mathbf{v})(\varphi(\mathbf{v}) \Leftrightarrow \psi_1(\mathbf{v})),$$
$$HW(\mathfrak{M}) \models (\forall \mathbf{v})(\neg\varphi(\mathbf{v}) \Leftrightarrow \psi_2(\mathbf{v})),$$

т.е. каждая Δ_0^**-формула и её отрицание определимы* Σ*-формулами без добавленных новых операторов*.

Из этой теоремы получаем важное следствие о консервативности полученного расширения на Σ-формулах.

С л е д с т в и е. *Для любой* $\Sigma(B\text{-list})(\mathfrak{M})$-формулы φ *существует* Σ-формула ψ *такая, что*

$$HW(\mathfrak{M}) \models (\forall \mathbf{v})(\varphi(\mathbf{v}) \Leftrightarrow \psi(\mathbf{v})).$$

Т е о р е м а 5. 1) *Для любого полиномиального* P-, PSPACE-*представления* $HW(\mathfrak{M})$ *с полиномиальной функцией* P-, PSPACE-*сложности* r *и* $\Delta_0(\mathfrak{I})$*-формулы* φ *существует полиномиальный* P-, PSPACE-*алгоритм проверки истинности этой формулы, т.е.* $HW(\mathfrak{M}) \models$ $\models \varphi(\mathbf{v})$ *на любом списке* $\mathbf{v}$ *из модели* $HW(\mathfrak{M})$.

2) *Для любого полиномиального* P-, PSPACE-*представления модели* $HW(\mathfrak{M})$ *с полиномиальной* P-, PSPACE-*функцией* r *и для любого* $\Delta_0(\mathfrak{I})$*-терма* $t(\mathbf{v})$ *определённая этим термом функция на модели* $HW(\mathfrak{M})$ *будет полиномиальной* P-, PSPACE-сложности.

Заметим, что при таком обогащении оказывается справедлив следующий результат.

Т е о р е м а 6. *Для любого полиномиального* P-, PSPACE-*представления модели* $HW(\mathfrak{M})$ *с полиномиальной функцией* r *и для любой полиномиально вычислимой как по пространству, так и по времени всюду определённой* P-, PSPACE-*функции на натуральных числах* Nat *существует* $\Delta_0(\mathfrak{I})$*-терм* $t(\mathbf{v})$ *такой, что определённая этим термом функция на модели* $HW(\mathfrak{M})$ *в точности определяет данную функцию на базисной модели* $\mathfrak{N}$*, в качестве которой взята стандартная модель арифметики со стандартным полиномиальным представлением*.

П р о б л е м а. При каких условиях для полиномиального P-, PSPACE- представления модели $HW(\mathfrak{M})$ любая полиномиально P-, PSPACE-вычислимая и Σ-определимая функция в $HW(\mathfrak{M})$ может быть определена $\Delta_0(\mathfrak{I})$-термом $t(\mathbf{v})$.

Источник финансирования. Исследование выполнено за счёт гранта Российского научного фонда (проект 17–11–01176).

СПИСОК ЛИТЕРАТУРЫ

1. *Гончаров С.С., Свириденко Д.И.* // Вычисл. системы. 1985. В. 107. С. 3–29.
2. *Goncharov S.S., Sviridenko D.I.* // Lect. Notes in Comput. Sci. 1986. V. 215. P. 169–179.
3. *Ershov Yu.L., Goncharov S.S., Sviridenko D.I.* Inform. proces., Proc. IFIP 10-th World Comput. Congr. Dublin, 1986. V. 10. P. 1113–1120.
4. *Гончаров С.С., Свириденко Д.И.* // ДАН. 1986. Т. 289. № 6. С. 1324–1328.
5. *Ershov Yu.L., Goncharov S.S., Sviridenko D.I.* // Lect. Notes in Comput. Sci. 1987. V. 278. P. 116–122.
6. *Ершов Ю.Л.* // ДАН. 1983. Т. 270. № 4. С. 786–788.
7. *Ершов Ю.Л.* // ДАН. 1983. Т. 273. № 5. С. 1045–1048.
8. *Ershov Yu.L.* Definability and Computability. Siberian School of Algebra and Logic. N.Y.: Consultants Bureau, 1996. 264 p.
9. *Barwise J.* Admissible Sets and Structures. Berlin: Springer, 1975. 394 p.
10. *Arora S., Barak B.* Computational Complexity. A Modern Approach. Cambridge: Cambridge Univ. Press, 2009. 605 p.
11. *Papadimitriou Ch.* Computational Complexity. Reding: Addison-Wesley, 1994. 523 p.
12. *Ospichev S.S., Ponomarev D.* // Siberian Electronic Math. Reps. 2018. V. 15. P. 987–995.
13. *Гончаров С.С.* // Сиб. мат. журн. 2017. Т. 58. № 5. С. 1026–1034.

LOGICAL LANGUAGE OF DESCRIPTION OF POLYNOMIAL COMPUTING

Academician of the RAS S. S. Goncharov, D. I. Sviridenko

Recieved October 16, 2018

An expansion of the term concept and, accordingly, the formula due to new operators were constructed. These extensions of the language preserve the expressiveness of Σ-formulas, and at the level of Δ_0-formulas and terms, they ensure the polynomiality of algorithms for calculating the value of a term and the truth of a Δ_0-formula.

Keywords: logic, bounded quantifiers, hereditarily finite list, computable model, term, conditional term, recursive term, formula, Σ-formula, Δ_0-formula, polynomial algorithms, polynomial model, semantic programming.