УДК 510.6

А.П. Пынько

# Приложение логического программирования к многозначной логике с относительной истиной

Предложены процедуры автоматического вывода в пропозициональных многозначных логиках с относительной истиной (имеющих несколько множеств выделенных значений) на основе многостороннего секвенциального дедуктивного аппарата, реализуемого методами логического программирования на языке ПРОЛОГ и позволяющего охватить все конечнозначные логики.

The procedures are suggested of automatic deduction in propositional many-valued logics with relative truth (that is, logics, which have several sets of distinguished values) based on the many-sided sequent deductive apparatus, which is implemented by logic programming methods in the PROLOG language and enables one to cover all finitely-valued logics.

Запропоновано процедури автоматичного виведення в пропозиційних багатозначних логіках з відносною істиною (таких, що мають кілька множин вилучених значень) на засадах багатобічного секвенційного дедуктивного апарата, що реалізується методами логічного програмування на мові ПРОЛОГ та надає можливість охопити усі скінченнозначні логіки.

**Введение**. Как известно, выбор типа и языка программирования для решения конкретной вычислительной задачи определяется ее спецификой. Для задач в области логики естественно использование логического программирования, воплощенного в языке ПРОЛОГ, особенно в тех случаях, когда необходимо реализовывать рекурсивные процедуры [1]. В данной статье ставится задача построения логических программ, работающих с дедуктивными аспектами многозначных логик. Более конкретно рассмотрены многозначные логики и соответствующие многосторонние секвенциальные исчисления, введенные в разд. 5 работы [2]. Многозначные логики указанной работы представляют собой наиболее широкий класс логических систем, для которых построены многосторонние секвенциальные исчисления, обобщающие исчисление Генцена $LK$ [3, 4] для классической логики и состоящие из атомарных аксиом и правил введения связок. Они содержат в качестве собственных частных случаев как первоначальные многосторонние исчисления, введенные в независимых работах [5, 6] для произвольных конечнозначных абстрактных алгебр, где они назывались логиками, так и более современные подходы работ [7] (в эквивалентном дуальном формализме табличных исчислений) и [8] (каким именно образом все перечисленные более ранние универсальные подходы вкладываются в подход работы [2], показаны в замечании 1 данной статьи. Как принято, для упрощения формализма оперируем не многосторонними секвенциями, а мультимножествами (конечными множествами с повторениями) меченых формул, где метки соответствуют сторонам. Согласно основному результату разд. 4 работы [2] всякое пропозициональное меченое секвенциальное исчисление с правилами утончения и сокращения, допускающее одновременное утончение посылок и заключений его правил одной и той же меченой формулой, есть полным и корректным по отношению к семантике в виде класса того, что называется здесь многозначными логиками с относительной истиной. Такие логики представляют собой обобщения обычных многозначных логик с единственным множеством выделенных значений (скажем, абсолютной истиной) и фактически имеют несколько множеств выделенных значений, соответствующих меткам, которые они интерпретируют. При этом метки выступают в роли мнений, исходя из которых одни и те же высказывания приобретают, вообще говоря, разные логические оценки («истинно» или «ложно») при одних и тех же интерпретациях пропозициональных букв истинностными значениями.

ПРОЛОГ-программы, разрабатываемые в настоящей статье, предназначены для верификации выводимости секвенций в секвенциальных исчислениях работы [2] для многозначных логик с относительной истиной (т.е. истинности секвенций в соответствующих логиках), а в случае положительного ответа – еще и построения вывода проверяемой секвенции в рассматриваемом исчислении. Идейно данная работа базируется на статье [9], где были предложены ПРОЛОГ-программы для решения аналогичной задачи, но для стандартных двухсторонних секвенций и многозначных логик с абсолютной истиной, имеющих определитель равенства. Использование определителя равенства (понятие введено в [10]), в отличие от работ [2, 5–8], позволило применить указанный подход не только к двухзначным логикам. Однако, как было доказано в работе [11], далеко не все многозначные логики с абсолютной истиной имеют определитель равенства. Поэтому, для того, чтобы применять логическое программирование к произвольным конечнозначным логикам, необходима разработка, представленная в настоящей статье.

Прежде чем перейти к изложению основного материала, целесообразно обсудить вопрос эффективности ПРОЛОГ-программ работы [9]. Вычислительные эксперименты, проведенные с ними, дали следующие результаты. Проверялись секвенции длиной порядка 100 символов пропозициональных букв и связок, причем среди них имелись как истинные, так и ложные секвенции. Время выполнения соответствующих запросов указанными ПРОЛОГ-программами исчислялось секундами, а время ввода таких запросов – десятками минут. Таким образом, время ответа на вопрос более, чем на два порядка, превышает время ввода самого вопроса. При использовании вспомогательных ПРОЛОГ-программ непосредственной семантической проверки истинности секвенций время вычислений возрастало примерно на порядок. Поскольку подходы работ [2, 9] совпадают на двухзначных логиках, а эксперименты проводились, в том числе, и для классической логики с использованием исчисления Генцена $LK$, результаты экспериментов показывают целесообразность распространения подхода работы [9] на секвенциальные исчисления работы [2].

**Основные понятия и обозначения**

Относительно алгебраических понятий и обозначений автор следует в основном обычным соглашениям (например, [12]). Чтобы унифицировать обозначения, если не оговорено иначе, алгебры будут обозначаться жирными курсивными заглавными латинскими буквами, а их носители – соответствующими светлыми буквами.

(*Абстрактный*) *пропозициональный язык* – это произвольная алгебраическая сигнатура $L$, символы которой, будучи функциональными, рассматриваются в качестве (*пропозициональных*) *связок*. Зафиксируем счетное множество $V = \{p_i\}_{i \geq 0}$, элементы которого будем называть *пропозициональными буквами* или *переменными*. Абсолютно свободная $L$-алгебра с образующими из $V$ обозначается $\boldsymbol{F}$. Элементы $F$ называются (*пропозициональными*) *формулами* (фактически, это $L$-термы с переменными из $V$). Если дана $L$-алгебра $\boldsymbol{A}$, то *интерпретацией в* $\boldsymbol{A}$ называется всякий гомоморфизм из $\boldsymbol{F}$ в $\boldsymbol{A}$. (Как обычно, интерпретации в алгебрах отождествляются с их ограничениями на множество $V$ или даже на его подмножества, состоящие из «существенных» в рассматриваемом контексте переменных. При этом иногда будем использовать такие стандартные обозначения оценок, как $[w_i / a_i]_{i < n}$, где $n \geq 0$, $w_0, \ldots, w_{n-1}$ – попарно различные переменные, а $a_0, \ldots, a_{n-1}$ – элементы рассматриваемой алгебры. В этом случае значения оценок на формулах записываются постфиксно). Оценки в $\boldsymbol{F}$ называются *подстановками*.

Зафиксируем непустое конечное множество унарных предикатных символов $\mathfrak{R}$, которые будем называть *мнениями* или *метками*. (*Многозначная*) *логика* (*с относительной истиной*) – это произвольная алгебраическая система $M$ сигнатуры $L \cup \mathfrak{R}$. (Такие логики соответствуют *меченым матрицам* [2].) Алгебру $M|L$ (обеднение $M$ на $L$) будем называть *алгеброй* многозначной логики $M$, а элементы ее носителя – (*исинностными*) *значениями* $M$. Будем называть $M$ *дискриминаторной*, если для любых ее различных значений $a$ и $b$ существует такое мнение $v$, что $a \in v^M$ и $b \notin v^M$, и для каждого мнения $v$, $v^M$ – непустое собственное подмножество носителя $M$. В этом случае $M$ – конечнозначна и имеет более одного истинностного значения. (Дискриминаторные логики соответствуют *сингулярным* меченым матрицам [2]).

Атомарные формулы сигнатуры $L \cup \Re$ с переменными из $V$ будем называть *мечеными* (*пропозициональными*) *формулами*.

(*Конечное*) *мультиподмножество* множества $S$ – это произвольная функция $\Delta : S \to \omega$, такая, что множество $\mathrm{Car}(\Delta) := \Delta^{-1}[\omega \backslash \{0\}]$, называемое *носителем* $\Delta$, конечно. Бинарные операции суммы $\oplus$ и разности $\backslash$ мультимножеств определяются покомпонентно через бинарные операции $\langle m,n \rangle \mapsto m+n$ и $\langle m,n \rangle \mapsto \max(m-n,0)$ на $\omega$. Если даны мультиподмножество $\Delta$ множества $S$ и одноместная операция $f$ на $S$, *f-мультиобраз* $\Delta$ определяется как мультиподмножество $f * \Delta$ множества $S$ следующим образом. Для каждого $a \in S$, полагаем $(f * \Delta)(a) := \sum\{\Delta)\ (b) \mid b \in \mathrm{Car}(\Delta),\ f(b) = a\}$. (Таким образом, $f$-мультиобраз сохраняет число элементов прообраза, даже если $f$ – неинъективна.) Мультимножества с областью значений $\subseteq \{0, 1\}$ отождествляются с их носителями. (Тем самым подмножества $S$ отождествляются с их характеристическими функциями.)

(*Меченая*) *секвенция* – это произвольное мультимножество $\Delta$ меченых формул. Семантически она интерпретируется формулой первого порядка $\vee\Delta$, что позволяет перенести понятие истинности (при интерпретациях в алгебрах отдельных многозначных логик или в множествах многозначных логик) из логики первого порядка на секвенции.

*Секвенциальная подстановка* [2] – это произвольная пара вида $\boldsymbol{\sigma} = \langle \sigma, \Sigma \rangle$, где $\sigma$ – подстановка, а $\Sigma$ – секвенция. (Чтобы унифицировать обозначения, если не оговорено иначе, секвенциальные подстановки обозначаются полужирными строчными греческими буквами, их первые компоненты – теми же нежирными буквами, а вторые – соответствующими заглавными.) Ее действие на произвольную секвенцию $\Delta$ определяется следующим образом: $\boldsymbol{\sigma}\Delta := (\sigma * \Delta) \oplus \Sigma$. Для множества секвенций $X$ полагаем $\boldsymbol{\sigma} X := \{\boldsymbol{\sigma}\Delta \mid \Delta \in X\}$. Секвенциальные подстановки образуют моноид с единицей $\boldsymbol{\iota} := \langle \iota, \varnothing \rangle$, где $\iota$ – тождественная подстановка, и произведением $\otimes$, определяемым следующим образом: $\boldsymbol{\sigma} \otimes \boldsymbol{\theta} := \langle \sigma \circ \theta, \Sigma \oplus \boldsymbol{\sigma}\Theta \rangle$. При этом $\boldsymbol{\iota}\Delta = \Delta$ и $(\boldsymbol{\sigma} \otimes \boldsymbol{\theta})\Delta = \boldsymbol{\sigma}(\boldsymbol{\theta}\Delta)$.

Понятия (*секвенциального*) *правила*, его *посылки* и *заключения*, *обратного правила*, (*секвенциальной*) *аксиомы* и (*секвенциального*) *исчисления* определяются стандартно. При этом правило $R = X/\Phi$ интерпретируется формулой $\wedge X \to \Phi$, что распространяет понятие истинности и на секвенциальные правила. Действие секвенциальных подстановок на правила распространяется очевидным образом $\boldsymbol{\sigma} R := \dfrac{\boldsymbol{\sigma} X}{\boldsymbol{\sigma}\Phi}$, а правила данного вида называются (*подстановочными*) *примерами* $R$. При этом истинность правил влечет истинность примеров.

Понятия *вывода в* исчислении и секвенции, *выводимой в* исчислении, определяются почти стандартно за исключением того, что при построении выводов используются не только исходные правила исчисления, но и все их примеры.

**Основные результаты**

Зафиксируем конечный пропозициональный язык $L$ и дискриминаторную логику $M$. Положим $\boldsymbol{A} := M|L$. Зафиксируем непустое множество $\{\boldsymbol{A}_j\}_{j<m}$ подалгебр $\boldsymbol{A}$, такое, что элементы $\{A_j\}_{j<m}$ попарно несравнимы (по включению). Для каждого $j<0$ положим $M_j := M|A_j$ (ограничение $M$ на подмножество $A_j$). И, наконец, положим $\boldsymbol{M} := \{M_j\}_{j<m}$. (В частном случае, когда $m = 1$ и $A_0 = A$, имеем $\boldsymbol{M} := \{M\}$. Этот случай будем называть *классическим*.)

**Замечание 1**. Для любой неодноэлементной конечной $L$-алгебры $\boldsymbol{B}$ будем рассматривать ее элементы в качестве меток. Получается дискриминаторная многозначная логика с относительной истиной $N$ сигнатуры $L \cup B$, определяемая следующим образом. Положим $N|L := \boldsymbol{B}$ и $v^N :=$ $:= \{v\}$ ($v^N := B \setminus \{v\}$) для всех $v \in B$. В классическом случае она соответствует подходу работы [5] (соответственно [6]). Кроме того, имеется дискриминаторная многозначная логика с относительной истиной $K$ сигнатуры $L \cup (2^B \setminus \{\varnothing, B\})$, определяемая следующим образом: $K|L :=$ $:= \boldsymbol{B}$ и $v^K := v$ для всех $v \in 2^B \setminus \{\varnothing, B\}$. В классическом случае она соответствует подходу работы [7] (поскольку множество меток замкнуто относительно операции теоретико-множественного дополнения, дуальный табличный формализм упомянутой работы эквивалентен секвенциальному с той же логикой). Общий классический случай изучен в работе [8].

Из аргументации замечания 1, в частности вытекает

**Замечание 2**. Любая неодноэлементная конечнозначная логика $N$ с носителем $B$ преобразуема в дискриминаторную $K$ отбрасыванием всех меток $v$ с $v^N \in \{\varnothing, B\}$ и добавлением новых меток из $B$ с интерпретацией $v^K := \{v\}$ для всех $v \in B$.

Пусть $W \subseteq V$. Под *$W$-атомом* понимается всякий элемент множества $At(W) := \{v(w) \mid v \in \mathfrak{R}$, $w \in W$. $V$-атомы называются просто *атомами*. Секвенция называется *простой* [($W$–) *атомарной*], если она – множество [соответственно состоит из ($W$–) атомов].

**Определение 1**. *Секвенциальная таблица для* $\boldsymbol{M}$ – это всякое отображение $T$ множества $L \times \mathfrak{R}$ в множество конечных множеств простых атомарных секвенций, таких, что для всех $n \geq 0$, всякой $n$-местной $\mu \in L$ и каждого $v \in \mathfrak{R}$, $T(\mu, v)$ состоит из $\{p_i\}_{i<n}$-атомарных секвенций, таких, что следующее правило $R_T(\mu, \nu) := \dfrac{T(\mu, \nu)}{\left\{\nu\left(\mu\left(p_i\right)_{i<n}\right)\right\}}$ и все, обратные ему, истинны в $\boldsymbol{M}$.

**Предложение 1**. Существует секвенциальная таблица для $\boldsymbol{M}$.

**Доказательство**. Возьмем произвольные $a \in$ $\in A$ и $i < n$. Множество всех $\{p_i\}$-атомов, ложных при интерпретации $[p_i/a]$ в $M$, обозначим $F_i(a)$. Заметим, что

$$a = b \Leftrightarrow \forall \gamma \in F_i(a) : \gamma \tag{1}$$

ложен при интерпретации $[p_i/b]$ в $M$ для всех $b \in A$.

Рассмотрим произвольные $n \geq 0$, $n$-местную $\mu \in L$ и $\nu \in \mathfrak{R}$. С учетом (1), видим, что $T(\mu, \nu) := \left\{\bigcup_{i<n} \mathrm{F}_i\left(a_i\right) \mid \left\langle a_i\right\rangle_{i<n} \in A^n, \mu^M\left(a_i\right)_{i<n} \notin \nu^M\right\}$ – конечное множество простых $\left\{p_i\right\}_{i<n}$-атомарных секвенций, таких, что правило $\dfrac{T(\mu, \nu)}{\left\{\nu\left(\mu\left(p_i\right)_{i<n}\right)\right\}}$ и все, обратные ему, истинны в $\{M\}$, а следовательно, и в $\boldsymbol{M}$. Таким образом, получаем секвенциальную таблицу $T$ для $\boldsymbol{M}$. □

Заметим, что доказательство предложения 1 не только конструктивно, но и дает аналитическое выражение (а значит, и эффективную процедуру вычисления) секвенциальной таблицы.

Подстановки в $V$ называются *атомарными*. Для двух простых атомарных секвенций $\Gamma$ и $\Delta$, полагаем $\Gamma \prec \Delta$, если существует такая инъективная атомарная подстановка $\sigma$, что $\sigma[\Gamma] \subseteq \Delta$. Бинарное отношение $\prec$ есть квазиупорядочение на множестве всех простых секвенций. Поэтому $\equiv := \prec \cap \succ$ – отношение эквивалентности на этом множестве, относительно которого две секвенции эквивалентны, только если они получаются одна из другой взаимнооднозначным переименованием переменных.

Конечное множество всех простых $\left\{p_j\right\}_{j<m}$– атомарных секвенций, истинных в $\boldsymbol{M}$, обозначим $\overline{Ax}$. Для каждого элемента $K$ фактор-множества $\overline{Ax}/\equiv$ выберем произвольную секвенцию $\Theta_K \in K$. Тогда $\prec$ будет частичным упорядочением на конечном множестве $\overline{\overline{Ax}} := \{\Theta_K \mid K \in$ $\in \overline{Ax}/\equiv\}$ простых атомарных секвенций. Конечное множество всех минимальных (по $\prec$) элементов $\overline{\overline{Ax}}$ обозначим $Ax$.

Зафиксируем секвенциальную таблицу $T$ для $\boldsymbol{M}$.

**Определение 2**. Конечное исчисление $C$ состоит из аксиом $Ax$ и для каждого $n \geq 0$ каждой $n$-местной $\mu \in L$ и каждого $v \in \mathfrak{R}$ правила $R_T(\mu, v)$.

**Замечание 3**. Любой пример любого правила $C$ и все правила, обратные примеру, истинны в $\boldsymbol{M}$.

Покажем, как дедуктивный формализм, принятый в настоящей статье, вписывается в синтаксис логического программирования. Пропозициональные буквы будут интерпретироваться функциональными константами. Тем самым пропозициональные формулы будут представляться основными термами. Представляя мнения одноместными функциональными (а не предикатными) символами, автор интерпретирует основными термами и меченые формулы. Представляя мультимножества списками (т.е. конечными последовательностями), тем самым будем интерпретировать секвенции списками основных термов. Для формального представления деревьев вывода вводим трехместный функциональный символ `tree`. Понятие (*канонического*) *дерева вывода* секвенции в $C$ определяется индуктивно следующим образом. Основной терм вида $\mathrm{tree}(\Delta,\sigma[\Xi],[t_i]_{i<n})$, где $n\in\omega$, а $\Delta$ – секвенция, называется (*каноническим*) *деревом вывода* $\Delta$ в $C$, если для каждого $i<n$, $t_i$ – дерево вывода некоторой секвенции $s_i$ в $C$, и существуют такие секвенциальная подстановка $\sigma$ и правило $R=\dfrac{X}{\Xi}\in C$, что $\sigma X=\{s_i\}_{i<n}$ и $\sigma\Xi=\Delta$, причем, если $R\in Ax$, то $\sigma$ – атомарна и инъективна, а $\sum$ – атомарна. (Понятно, что первые два аргумента содержат полную информацию о паре $(\sigma, R)$, поскольку правила $C$ однозначно определяются их заключениями и, в силу инъективности $\sigma$ при $R\in Ax$, одноэлементности $\Xi$ при $R\notin Ax$ и простоты заключения любого правила $C$, $\sigma*\Xi=\sigma[\Xi]$, и поэтому $\sum=\Delta\setminus\sigma[\Xi]$, а секвенция $\sigma[\Xi]$ однозначно определяет пару $(\sigma,\Xi)$, потому что множество $Ax$ служит скелетом относительно эквивалентности $\equiv$.) Кроме того, вводится трехместный предикатный символ `seq`. Запрос вида: $-$`seq` $(\Delta,\Xi,U)$, где $\Delta(\Xi)$– (атомарная) секвенция, а $U$ – произвольная переменная ПРОЛОГа, называется *каноническим*. При этом $\Delta\oplus\Xi$ называем *характеристической секвенцией*, а $U$ – *переменной* рассматриваемого запроса. В качестве встроенных предикатов *SWI*-ПРОЛОГа (см. *http://www.swi-prolog.org/*), используется нульместный предикат ! *отсечения*, двухместный предикат `member` (в случае, когда второй аргумент – список основных термов, данный предикат ищет основной частный случай первого аргумента, который есть элементом второго аргумента) и двухместный предикат `memberchk` (в случае, когда второй аргумент – список основных термов, а первый – основной терм, данный предикат проверяет, есть ли первый аргумент элементом второго, но, в отличие от предиката `member`, предикат `memberchk` не оставляет точек возврата).

**Определение 3**. ПРОЛОГ-программа $P$ состоит из следующей последовательности правил и фактов:

1) для каждой $\Xi\in Ax$, следующее правило, где $\Xi\left[p_j/\mathrm{P}j\right]_{j<m}=\left\{\psi_q\right\}_{q<r}$ и, для всех $q<r$, $V_q$ – переменная ПРОЛОГа, входящая в $\psi_q$, $\eta_q$ – пустое слово, а $\chi_q=,\mathtt{not}(\mathtt{member}(V_q,\left\{V_u\right\}_{u<q}))$, если $V_q\notin\left\{V_u\right\}_{u<q}$, и $\eta_q=\mathrm{chk}$, а $\chi_q$ – пустое слово, – в противном случае

$$\mathtt{seq}([\,],Y,\mathtt{tree}(X,[\psi_q]_{q<r},[\,])):-$$
$$\left\langle\mathtt{member}\,\eta_q(\psi_q,Y)\chi_q,\right\rangle_{q<r}!,\mathtt{append\_rev\ erse}(Y,[\,],X).$$

2) Для каждого $n\geq 0$, каждой $n$-местной $\mu\in L$ и каждого $\nu\in\Re$, следующее правило, обозначаемое $\boldsymbol{R}_T(\mu,\nu)$, где $T(\mu,\nu)\left[p_i/Pi\right]_{i<n}=\left\{\Omega_k\right\}_{k<l}$:

$$\mathtt{seq}([\nu(\mu(Pi)_{i<n})|Y],Z,\mathtt{tree}(X,[\nu(\mu(Pi)_{i<n})],$$
$$[Tk]_{k<l}):-!,\left\langle\mathtt{seq}([\Omega_k|Y],Z,Tk),\right\rangle_{k<l}$$
$$\mathtt{append\_rev\ erse}(Z,[\nu(\mu(Pi)_{i<n})|Y],X).$$

3) $\mathtt{seq}([L|Y],Z,T):-\mathtt{seq}(Y,[L|Z],T)$.

4) $\mathtt{append\_rev\ erse}([\,],X,X):-!.$

$$\mathtt{append\_rev\ erse}([X|Y],Z,W):$$
$$-\mathtt{append\_r\ everse}(Y,[X|Z],W).$$

*Степень* $\partial(\varphi)$ формулы $\varphi$ определяется индукцией по числу вхождений связок в $\varphi$ следующим образом:

- для всякой $w\in V$, полагаем $\partial(w):=1$;

• для всякого $n \geq 0$, каждой $n$-местной $\mu \in L$ и всех $\langle \varphi_i \rangle_{i<n} \in F^n$, полагаем $\partial\left(\mu\left(\varphi_i\right)_{i<n}\right) := 1 + |\Re| \cdot \sum_{i<n} \partial\left(\varphi_i\right)$.

Отметим, что для любой формулы $\varphi$, $\partial(\varphi)>0$. Далее, для всякого $v \in \Re$ и каждой $\varphi \in F$, полагаем $\partial(v(\varphi)) := \partial(\varphi)$. Наконец, для произвольной секвенции $\Gamma$, полагаем $\partial(\Gamma) := \sum\{\Gamma(\Phi) \cdot \partial(\Phi) \mid \Phi \in \mathrm{Car}(\Gamma)\}$.

**Лемма 1**. Канонический запрос получает в программе $P$ положительный ответ с решением «переменная запроса = дерево вывода характеристической секвенции запроса в $C$», если характеристическая секвенция запроса истинна в $\boldsymbol{M}$, и отрицательный ответ – в противном случае.

**Доказательство**. Индукцией по *степени* $\partial(\Delta)$ заданного канонического запроса $:\!-\mathrm{seq}(\Delta,\Xi,U)$.

Предположим, что $\partial(\Delta) = 0$. Тогда $\Delta = [\,]$. Поэтому, рассматриваемый запрос сопоставим только с заголовками правил п. 1) определения 3. Если существуют такие $\Theta \in Ax$ и инъективная атомарная подстановка $\sigma$, что $\sigma[\Theta] \subseteq \mathrm{Car}(\Xi)$, то данный запрос получает в $P$ положительный ответ с решением $U = \mathtt{tree}(\Xi,\sigma[\Theta],[\,])$, и тогда $\Xi = \sigma\Theta$, где $\Sigma := \Xi \setminus \sigma[\Theta]$, поскольку $\sigma * \Theta = \sigma[\Theta]$, а $\mathtt{tree}(\Xi,\sigma[\Theta],[\,])$ – дерево вывода $\Xi$ в $C$. В противном случае, запрос получает в $P$ отрицательный ответ, поскольку $\Xi$ атомарна. Предположим теперь, что $\Xi$ истинна в $\boldsymbol{M}$. От противного докажем, что для всех $j < m$, существует такая $w \in V$, что секвенция $\mathrm{Car}(\Xi) \cap At(\{w\})$ истинна в $\{M_j\}$. Предположим, что существует такое $j < m$, что для всех $w \in V$, существует интерпретация $h_w$ в $\boldsymbol{A}_j$, при которой $\mathrm{Car}(\Xi) \cap At(\{w\})$ ложна в $M_j$. Определим интерпретацию $h$ в $\boldsymbol{A}_j$, положив $h(w) := h_w(w)$ для всех $w \in V$. Очевидно, что секвенция $\Xi$ ложна в $M_j$ при интерпретации $h$, поскольку она атомарна. Это противоречит ее истинности в $\boldsymbol{M}$. Таким образом, для всех $j < m$, существует такая $w_j \in V$, что простая $\{w_j\}$-атомарная секвенция $\Gamma_j := (\mathrm{Car}(\Xi) \cap At(\{w\}))$ истинна в $\{M_j\}$. Поэтому простая $\{w_j\}_{j<m}$-атомарная секвенция $\Omega := \bigcup_{j<m} \Gamma_j$ истинна в $\boldsymbol{M}$. Поскольку $|\{w_j\}_{j<m}| \leq m$, существует инъекция $\gamma : \{w_j\}_{j<m} \to \{p_j\}_{j<m}$. Тогда $\gamma[\Omega] \in \overline{Ax}$. Поэтому существует такая $\Theta \in Ax$, что $\Theta \prec \gamma[\Omega]$. Кроме того, $\gamma^{-1}[\gamma[\Omega]] = \Omega \subseteq \mathrm{Car}(\Xi)$, а $\gamma^{-1}$ – инъекция. Поэтому $\gamma[\Omega] \prec \mathrm{Car}(\Xi)$. Тогда $\Theta \prec \mathrm{Car}(\Xi)$. Таким образом замечание 3 завершает рассмотрение данного случая.

Теперь предположим, что $\partial(\Delta) > 0$. Тогда $\Delta \neq [\,]$. Поэтому $\Delta = [\Phi \mid \Lambda]$.

Сначала предположим, что $\Phi$ – атом. Тогда правило п. 3) определения 3 – единственное правило программы $P$, заголовок которого сопоставим с рассматриваемым каноническим запросом. Поэтому исходный запрос получает в программе $P$ отрицательный ответ (положительный ответ с некоторым решением) тогда и только тогда, когда производный канонический запрос $:\!-\mathtt{seq}(\Lambda,[\Phi|\Xi],U)$ получает в программе $P$ отрицательный ответ (положительный ответ с тем же решением). Заметим, что оба запроса имеют одну и ту же характеристическую секвенцию. Кроме того, поскольку $\partial(\Phi)>0$, степень исходного запроса будет больше степени производного. Таким образом, индукционное предположение завершает рассмотрение данного частного случая.

Теперь предположим, что $\Phi$ не есть атомом. Тогда существуют такие $n \geq 0$, $n$-местная $\mu \in L$, $\langle \varphi_i \rangle_{i<n} \in F^n$ и $v \in \Re$, что $\Phi = v(\mu(\varphi_i)_{i<n})$. Правило $\boldsymbol{R}_T(\mu, v)$ из п. 2) определения 3 – первое правило программы $P$, заголовок которого сопоставим с рассматриваемым каноническим запросом, что приводит к проверке интерпретатором ПРОЛОГа посылок данного правила. Выполнение первой посылки – предиката отсечения ! – исключает использование альтернативных правил для исходного канонического запроса. Поэтому он получает положительный (отрицательный) ответ в программе $P$ тогда и только тогда, когда каждый (один) из производных запросов с предикатом `seq`, будучи каноническим, получает положительный (отрицательный)

ответ в программе $P$. Заметим, что степень исходного канонического запроса больше степени любого производного. Кроме того, в то время как характеристическая секвенция исходного канонического запроса есть заключением правила $\sigma R_T(\mu,\nu)$, где $\sigma := [p_i/\varphi_i]_{i<n}$ и $\Sigma := \Lambda \oplus \Xi$, множество характеристических секвенций всех производных канонических запросов совпадает с множеством посылок указанного правила. Тем самым индукционное предположение и замечание 3 завершают рассмотрение данного последнего частного случая, а вместе с ним и доказательство леммы 1. □

**Следствие 1**. Программа *P* завершаема на канонических запросах.

Как частный случай леммы 1 имеем:

**Теорема 1**. Пусть $\Delta$ – произвольная секвенция. Тогда запрос `:-seq(`$\Delta$`,[],T)` получает в программе *P* положительный ответ с решением «$T$ = дерево вывода $\Delta$ в $C$», если $\Delta$ истинна в ***M***, и отрицательный ответ – в противном случае.

Теорема 1 и замечание 3 непосредственно дают

**Следствие 2**. Пусть $\Delta$ – произвольная секвенция. Тогда следующие условия эквивалентны:

- запрос `:-seq(`$\Delta$`,[],`$T$`)` получает в программе *P* положительный ответ;
- $\Delta$ истинна в ***M***;
- $\Delta$ выводима в *C*;
- $\Delta$ имеет каноническое дерево вывода в *C*.

Существует естественное эквивалентное представление автономной программы *P* в виде объединения универсальной ПРОЛОГ-программы, выделяющей общие элементы из *P*, и базы данных на языке ПРОЛОГ, которая интерпретирует множество *Ax* и функцию *T*. Для этой цели введем новые предикаты: одноместный `axiom` и двухместный `table`. Кроме того, используем дополнительные встроенные предикаты: одноместный предикат отрицания `not`, одноместный предикат `ground` (проверяет, есть ли его аргумент основным термом), двухместный предикат `term_varia bles` (второй аргумент – множество переменных первого) и трехместный предикат `append` (если первые два аргумента – списки основных термов, а третий – переменная ПРОЛОГа, то выполнение запроса `:-append(`$L,M,U$`)` завершается положительным ответом с решением «$U$ = конкатенация $L$ и $M$»).

**Определение 4**. ПРОЛОГ-программа *UP* состоит из следующих предложений:

1) `seq([],X,tree(Y,A,[])):-axiom(A), subset_con text(A,X,[]),!,append_inv erse(X,[],Y).`

2) `seq([X|Y],Z,tree(W,[X],T)):-table(X, S),!, premises(Y,Z,S,T)append_inv erse(Z, [X|Y],W).`

3) предложение п. 3) определения 3;

4) `premises(Y,Z,[],[]):-!.`
`premises(Y,Z,[A|S],[B|T]):-append(A ,Y,X), seq(X,Z,B),premises(Y,Z,S,T).`

5) предложения п. 4) определения 3;

6) `subset_con text([],X,Y):-!.`
`subset_con text([X|Y],Z,L):- member_con text(X,Z,L,M),subset_con text(Y,Z,M).`

7) `member_con text(X,Y,L,L):- ground(X),!,memberchk(X,Y).`
`member_con text(X,Y,L,[V|L]):- term_var iables(X,[V]), member(X,Y),not(member(V,L).`

**Определение 5**. База данных *BD* на языке ПРОЛОГ состоит из следующих фактов:

- для каждой $\Theta \in Ax$, факт:

$$\texttt{axiom}\left(\Theta\left[p_j/\mathrm{P}j\right]_{j<m}\right);$$

- для каждого $n \geq 0$, каждой $n$-местной $\mu \in L$ и каждого $\nu \in \mathfrak{R}$, факт:

$$\texttt{table}(\nu(\mu(Pi)_{i<n}),T(\mu,\nu)\left[p_i/Pi\right]_{i<n}).$$

**Предложение 2**. Программы *P* и $UP \cup BD$ эквивалентны на канонических запросах.

**Доказательство**. Непосредственной проверкой, индукцией по степени канонических запросов по схеме доказательства леммы 1. □

В заключение основной части статьи следует отметить важность использования предиката отсечения `!` (в том числе неявного – в предикате `memberchk`, который превращается в предикат `member` после удаления предиката ! из его определения) для повышения эффективности предложенных программ. Поскольку любые примеры элементов $Ax$ истинны в $\boldsymbol{M}$, удаление каких-либо вхождений предиката ! из пп. 1) и 2) определений 3 и 4 даст по-прежнему корректные программы. Более того, такое удаление не влияет на процесс обработки интерпретатором ПРОЛОГа тех канонических запросов, характеристические секвенции которых истинны в $\boldsymbol{M}$. Но в случае ложной характеристической секвенции, интерпретатор будет анализировать слишком много ненужных альтернатив и, как следствие, время вычисления увеличится (как правило, значительно). В предложенном варианте, как только интерпретатор сталкивается с каноническим запросом нулевой степени с ложной характеристической секвенцией после перебора всего множества $Ax$ все точки возврата будут исчерпаны, и интерпретатор немедленно даст отрицательный ответ. Удаление отсечений в п. 4) определений 3, 4 и пп. 5) и 6) определения 4 существенно не влияет на время вычисления. Но отсечения пп. 1), 2) и 4) определений 3, 4 и пп. 5) и 6) определения 4 выполняют функцию *оптимизация остатка рекурсии* (см. разд. 11.2 в [1]) и, тем самым дают возможность существенно экономить память (стек), улучшить сборку мусора и избежать неоправданного переполнения стека, что приводит к немедленному завершению работы интерпретатора с ошибкой.

**Заключение**. Итак, теорема 1, предложение 2 и замечание 2 показывают, что данная работа решает поставленную задачу построения логических программ, анализирующих выводимость секвенций в исчислениях [2] для произвольных многозначных логик с относительной истиной.

Представленная разработка может быть использована в преподавательской деятельности при изучении логики и дискретной математики. Кроме того, предложенные ПРОЛОГ-программы можно рассматривать в качестве прототипов при реализации систем искусственного интеллекта (таких, как оболочки экспертных систем), базирующихся на многозначных логиках (например, таких, которые могут обрабатывать противоречивую и/или неполную информацию).

Хотя использование логик с относительной истиной не требует привлечения определителей равенства для того, чтобы охватить все многозначные логики, было бы целесообразно иметь единый подход, который включал бы работу [9] и представленную статью как частные случаи. В объединении двух векторов разработок и состоит основное дальнейшее развитие настоящей работы. Отметим, что такое сложение, но без логических программ уже сделано в работе [11].


1. *Стерлинг Л.*, *Шапиро Э.* Искусство программирования на языке ПРОЛОГ. – М.: Мир, 1990. – 235 с.
2. *Pynko A.P.* Semantics of multiplicative propositional signed sequent calculi with structural rules // J. of Multiple-Valued Logic and Soft Computing. – 2004. – **10**. – P. 339–362.
3. *Gentzen G.* Untersuchungen über das logische Schließen (I) // Mathematische Zeitschrift. – 1934. – **39**. – P. 176–210.
4. *Gentzen G.* // Ibid. (II). – P. 405–431.
5. *Schröter K.* Methoden zur Axiomatisierung beliebiger Aussagen- und Prädikatenkalküle // Zeitschrift für Mathematische Logik und Grundlagen der Mathematik. – 1955. – **1**. – P. 214–251.
6. *Rousseau G.* Sequents in many-valued logic (I) // Fundamenta Mathematicae. – 1967. – **60**. – P. 171–181.
7. *Hähnle R.* Automated deduction in multiple-valued logics. – Oxford: Oxford University Press, 1994. – 182 p.
8. *Labeled* calculi and finite-valued logics / M. Baaz, C. Fermüller, G. Salzer et al. // Studia Logica. – 1998. – **61**. – P. 7–33.
9. *Пынько А.П.* Процедуры вывода в секвенциальных исчислениях для конечнозначных логик с определителем равенства // Доп. НАН України. – 2007. – № 3. – С. 45–51.
10. *Pynko A.P.* Sequential calculi for many-valued logics with equality determinant // Bulletin of the Section of Logic. – 2004. – **33**. – P. 23–32.
11. *Pynko A.P.* Many-place sequent calculi for finitely-valued logics // Logica Universalis. – 2010. – **4**. – P. 41–66.
12. *Общая* алгебра: В 2 т. – М.: Наука, 1991. – Т. 2. – 480 с.


Поступила 05.06.2012


●