



Math-Net.Ru

All Russian mathematical portal

A. L. Rastsvetaev, A propositional logic of Boolean recursive programs in which predicate variables appear in conditions,

Mat. Zametki, 1990, Volume 48, Issue 3, 119–127

<https://www.mathnet.ru/eng/mzm3338>

Use of the all-Russian mathematical portal Math-Net.Ru implies that you have read and agreed to these terms of use

<https://www.mathnet.ru/eng/agreement>

Download details:

IP: 18.97.14.88

April 23, 2025, 14:40:59



О ПРОПОЗИЦИОНАЛЬНОЙ ЛОГИКЕ БУЛЕВЫХ РЕКУРСИВНЫХ ПРОГРАММ С ВХОЖДЕНИЕМ ПРЕДИКАТНЫХ ПЕРЕМЕННЫХ В УСЛОВИЯ

А. Л. Расцветаев

1. В настоящей работе определяется пропозициональная логика \mathcal{L} булевых рекурсивных схем программ, имеющих вхождения предикатных переменных в условия, и вводится отношение структурной интерпретируемости программных логик, обобщающее транслируемость, которое используется при установлении разрешимости логики \mathcal{L} и некоторых ее вариантов. Впервые схемы программ были рассмотрены в 1958 г. Ю. И. Яновым [1]. В 60-е и 70-е гг. рядом авторов изучались обобщения схем Янова, в частности, регулярные выражения Глушкова [2, 3]. В 1979 г. М. Фишером и Р. Ладнером метод фильтрации модальной логики был обобщен на пропозициональную динамическую логику PDL , в которой формализуются многие свойства регулярных схем программ 4 и позже распространен на PDL с константами [5]. В работе [6] объявлена экспоненциальная разрешающая процедура для логики PDL с детерминированными атомарными программами ($DPDL$) с константами. В [7] получена экспоненциальная процедура для логики $DPDL$ с заикливанием ($LDPDL$) с конструкцией обращения программ. Однако расширение класса схем контекстно-свободными рекурсивными схемами приводит к неразрешимости проблемы выполнимости. Рассматриваемая в настоящей работе управляющая структура булевых рекурсивных схем позволяет определить известную схему Патерсона — Хьюитта, неэквивалентную никакой стандартной схеме программ. Полученные результаты показывают, что эта управляющая структура позволяет также интерпретировать в \mathcal{L} постоянные и обратимые функциональные символы (но не перестановочные, см. [8]) и сама интерпретируется в логике $LDPDL$.

2. О п р е д е л е н и е. 1) Двухзначная пропозициональная логика есть пара (Γ, T) , где язык Γ есть множество *формул* и семантика T — множество *моделей* t , представляющих истинностно-значные функции $t: \Gamma \rightarrow \{И, Л\}$. Язык Γ строится индуктивно

из множества *атомарных формул* $\Gamma_0 \subseteq \Gamma$ с использованием классических пропозициональных связок \neg, \vee, \dots . Если $t\alpha = \mathbf{И}$, то говорят, что формула α выполняется в модели t , $t \models \alpha$. Формула α *общезначима*, $\models \alpha$, если $\forall t \in T$ ($t \models \alpha$) и *выполнима*, если $\not\models \neg \alpha$.

2) Эффективное отображение $\text{tr}: \Gamma \rightarrow \Gamma'$ называется *трансляцией* логики $L = (\Gamma, T)$ в логику $L' = (\Gamma', T)$, если $\forall \alpha \in \Gamma$ $\forall t \in T$ ($t \models \alpha \equiv t \models \text{tr}(\alpha)$); L *транслируется* в L' , если существует трансляция из L в L' ; L *сводится* к L' , если алгоритмическая проблема выполнимости для L сводится к проблеме выполнимости для L' .

Очевидно, что если L транслируется в L' , то L сводится к L' . Поскольку, однако, логики L и L' могут иметь различную семантику (например, класс моделей логики L может строго содержаться в классе моделей логики L'), то для установления ряда сводимостей ниже используется более общее понятие интерпретации.

О п р е д е л е н и е. Перестановочное с пропозициональными связками отображение i множества Γ формул логики $L = (\Gamma, T)$ в множество Γ' формул логики $L' = (\Gamma', T')$, называется *интерпретацией* L в L' , если $\forall \alpha \in \Gamma$ ($\forall t \in T \exists t' \in T'$: для каждой атомарной подформулы β формулы α выполнено $t \models \beta$ тогда и только тогда, когда $t' \models i\beta$) и обратное, $\forall t' \in T' \exists t \in T$: для каждой атомарной подформулы β выполнено $t \models \beta$ тогда и только тогда, когда $t' \models i\beta$).

Если L интерпретируется в L' , то L сводится к L' .

3. В то время как в языке динамической логики [9] термы вводятся независимо от формул, а понятие схемы программы есть обобщение понятия термина, ниже в терминах допускается вхождение формул, а схемы программ обобщают формулы.

Рассмотрим попарно непересекающиеся множества монадических символов — функциональных и предикатных констант \mathcal{F} и \mathcal{P} и предикатных переменных Π .

ФТ1. Индивидуальная переменная x есть функциональный терм.

ФТ2. Если τ — функциональный терм, а ε — функциональный символ, то $\varepsilon(\tau)$ — функциональный терм.

ЛТ1. Символы $\mathbf{И}$ и $\mathbf{Л}$ являются логическими терминами (представляющими соответственно истинностные значения «ИСТИНА» и «ЛОЖЬ»).

ЛТ2. Если α, β и β_1 — логические термы, то $\text{if } \alpha \text{ then } \beta \text{ else } \beta_1$ — логический терм (записывается в виде $(\alpha \mid \beta \mid \beta_1)$).

ЛТ3. Если τ — функциональный терм, а ε — предикатный символ, то $\varepsilon(\tau)$ — логический терм.

ЛТ4. Каждая формула является логическим термом.

Оп. Определением предикатной переменной называется выражение вида $\varepsilon(x) \leftarrow \tau(x)$, где $\tau(x)$ — логический терм.

Сх. Схемой программы (или \mathcal{L} — схемой) называется пара $\mathcal{Y} = (\mathcal{Y}^d, \mathcal{Y}^i)$, где \mathcal{Y}^d непустое конечное множество определений, а \mathcal{Y}^i — выделенная начальная переменная.

АФ1. Каждая схема является атомарной формулой.

АФ2. Если \mathcal{Y} — схема, то $\text{loop } \mathcal{Y}$ — атомарная формула.

Размер формулы есть число вхождений в нее символов.

Напомним, что интерпретацией сигнатуры $\langle \mathcal{F}, \mathcal{P} \rangle$ называется пара $I = (D, \gamma)$, где D — некоторое множество, называемое областью интерпретации, а γ — отображение, ставящее в соответствие каждому функциональному символу $f \in \mathcal{F}$ одноместную всюду определенную на D функцию $f^I: D \rightarrow D$ и каждому предикатному символу $p \in \mathcal{P}$ — одноместный всюду определенный на D предикат $p^I: D \rightarrow \{\mathbf{И}, \mathbf{Л}\}$. Модель есть пара $t = (I, d)$, где $d \in D$. Терм над D есть оцененный функциональный или логический терм (т. е. терм, в котором вместо индивидуальных переменных подставлены элементы D).

Вычислением схемы \mathcal{F} в модели $t = (I, d_0)$ называется последовательность s термов над D , удовлетворяющая следующим 3-м условиям.

I. Первый член последовательности есть терм $\mathcal{F}^i(d_0)$.

II. Если член c_i последовательности не заключительный, то следующий член c_{i+1} может быть получен из него

1) заменой термов всюду на их значения в соответствии с интерпретацией I до тех пор пока это возможно, по обычным правилам: $f(d) \rightarrow f^I(d)$; $p(d) \rightarrow p^I(d)$; $(\mathbf{И} \mid \tau \mid \dots)$, $(\mathbf{Л} \mid \dots \mid \tau) \rightarrow \tau$; $\alpha(d) \rightarrow \mathbf{И}$, если $(I, d) \models \alpha$ и $\mathbf{Л}$ — в противном случае (α — формула), и последующей

2) заменой самого левого вхождения предикатной переменной на правую часть своего определения в схеме \mathcal{F} .

III. Последовательность конечна тогда и только тогда, когда она содержит истинное значение.

Заметим, что если вычисление заканчивается, то терм в п. III, будет, в силу п. II, заключительным. Он называется значением вычисления.

Атомарная формула выполняется в модели, если (случай схемы) она имеет в этой модели конечное вычисление со значением $\mathbf{И}$ или (случай формулы заикливания) она имеет бесконечное вычисление.

Определение логики \mathcal{L} закончено. (Индивидуальная переменная x обычно опускается.) В \mathcal{L} транслируется логика $LDPDL$.

4. ТЕОРЕМА 1. Логика \mathcal{L} интерпретируется в логике $LDPDL$.

Доказательство. Расширим множество термов добавлением к индуктивному определению термов следующего пункта.

T. Если τ и τ' — логические термы, то $(\tau \cup \tau')$ — логический терм, с соответствующими правилами замены: $(\tau \cup \tau') \rightarrow \tau, \tau'$.

Введение такой конструкции недетерминированного выбора, очевидно, не расширяет класса свойств, выразимых формулами, но позволяет считать, что каждая переменная в схемах имеет не более одного определения. Без умаления общности можно, таким образом, предполагать, что термы, стоящие в правых частях определений \mathcal{L} -схем принадлежат одному из следующих типов:

$\mathbf{И}, \mathbf{Л}, p, pf, (\pi \cup \pi'), (\pi \mid \pi' \mid \pi'')$ или β (\mathcal{L} — формула), и что каждая переменная имеет ровно одно определение.

Достаточно определить отображение интерпретации i на атомарных формулах. Рассмотрим 2 возможности, которые могут представиться в соответствии с пп. АФ1 и АФ2.

1) Схема α . Поставим каждой предикатной переменной ρ схемы α во взаимнооднозначное соответствие предикатные константы

$$\mathbf{I}_\rho, \mathbf{I}\rho, \mathbf{H}_\rho, r_\rho^{\mathbf{I}}, r_\rho^{\mathbf{I}}, r_\rho^{\mathbf{H}},$$

не имеющие вхождений в исходную формулу, и положим

$$\chi_\rho \Leftrightarrow \bigwedge \{ \# \rho \equiv \# \tau_\rho \mid \# = \mathbf{I}, \mathbf{A}, \mathbf{H} \},$$

где формула $\# \tau_\rho$ определяется в соответствии с видом термина τ_ρ , стоящего в правой части определения ρ , следующим образом

$$\mathbf{I}\mathbf{I} = \mathbf{I}\mathbf{I} = \mathbf{I},$$

$$\mathbf{I}p = p, \quad \mathbf{I}p = \neg p,$$

$$\mathbf{I}\mathbf{I} = \mathbf{I}\mathbf{I} = \mathbf{H}\mathbf{I} = \mathbf{H}\mathbf{I} = \mathbf{H}p = \mathbf{H}\beta = \mathbf{I},$$

$$\mathbf{I}\beta = i\beta, \quad \mathbf{I}\beta = \neg i\beta \quad (p \in \mathcal{P}, \beta - \text{формула}),$$

$$\#(\pi f) = [f] \# \pi,$$

$$\#(\pi \cup \pi_1) = r_\rho^\# \& \# \pi \vee \neg r_\rho^\# \& \# \pi_1, \quad \# = \mathbf{I}, \mathbf{I}, \mathbf{H},$$

$$\#(\pi \mid \pi_1 \mid \pi_2) = r_\rho^\# \& \mathbf{I}\pi \& \# \pi_1 \vee \neg r_\rho^\# \& \mathbf{I}\pi \& \# \pi_2, \quad \# = \mathbf{I}, \mathbf{I},$$

$$\mathbf{H}(\pi \mid \pi_1 \mid \pi_2) = \mathbf{I}\pi \& \mathbf{H}\pi_1 \vee \mathbf{I}\pi \& \mathbf{H}\pi_2 \vee \mathbf{H}\pi.$$

Положим $\chi \Leftrightarrow [\mathcal{F}_\alpha^*] \wedge \{ \chi_\rho \mid \rho \in \Pi \text{ и } \rho \text{ имеет вхождение в } \alpha \}$, где $\mathcal{F}_\alpha \subseteq \mathcal{F}$ — множество функциональных констант, имеющих вхождения в схему α . (Конструкции логики $DPDL$ в \mathcal{L} используются как очевидные сокращения.)

Пусть α' — схема с множеством переменных $\{ \rho^\# \mid \rho \text{ входит в } \alpha, \# \mathbf{I}, \mathbf{I} \}$, начальной переменной $(\alpha')^i = (\alpha^i)^\mathbf{I}$ и определениями $\rho^\# \leftarrow \tau_\rho^\#$, где терм $\tau_\rho^\#$ следующим образом определяется в соответствии с термом τ_ρ из правой части определения ρ в схеме α :

$$\mathbf{I}^\# = \mathbf{I}^\# = p^\# = \beta^\# = \mathbf{I} \quad (p \in \mathcal{P}, \beta - \text{формула}),$$

$$(\pi f)^\# = \pi^\# f,$$

$$(\pi \cup \pi_1)^\# = (r_\rho^\# \mid \pi^\# \mid \pi_1^\#),$$

$$(\pi \mid \pi_1 \mid \pi_2)^\# = (r_\rho^\# \mid (\pi^\mathbf{I} \cup \pi_1^\#) \mid (\pi^\mathbf{I} \cup \pi_2^\#)), \quad \# = \mathbf{I}, \mathbf{I}.$$

Положим $i\alpha \Leftrightarrow (\alpha^i)^\mathbf{I} \& \neg \text{loop } \alpha' \& \chi$.

2) В случае атомарной формулы вида $\text{loop } \alpha$ положим

$$i \text{ loop } \alpha \Leftrightarrow (\alpha^i)^\mathbf{H} \& \chi.$$

Формулы $i\alpha$ и $i \text{ loop } \alpha$, очевидно, транслируются в $LDPDL$ — формулы линейного размера (как функции размера α). Следующее предложение показывает, что построенное отображение i есть отображение интерпретации.

Предложение 1. а) Пусть I — интерпретация с областью D и $(I, d) \models \alpha$. Тогда существует интерпретация I' с областью D такая, что $(I', d) \models i\alpha$; б) $\models i\alpha \rightarrow \alpha$.

Доказательство проводится индукцией по вычислениям, причем оба случая атомарных формул разбираются анало-

гично. Для схемы α , например, интерпретация I' отличается от I только быть может значениями предикатных констант $\{\# \pi, \tau_{\pi}^{\#} \mid \# = \mathbf{И}, \mathbf{Л}, \mathbf{Н}; \pi \text{ входит в } \alpha\}$, и выполнено $(I', d) \models i\alpha$ тогда и только тогда, когда предикатами $(\# \pi)^{I'}$ и $(\tau_{\pi}^{\#})^{I'}$ «описывается» некоторое заканчивающееся вычисление с схемы α в модели (I, d) с последним термом $\mathbf{И}$. При этом выполнение формул χ_{π} во всех элементах области D , достижимых из d посредством применения функций $f^{I'}, f \in \mathcal{F}_{\alpha}$, гарантирует «локальную правильность» вычисления, а выполнение формулы $\bigwedge \text{loop } \alpha'$ — его завершение.

Доказательство теоремы 1 закончено.

Из теоремы 1 и результата [7] о разрешимости логики $LDPDL$ за экспоненциальное время получаем

С л е д с т в и е. Алгоритмическая проблема выполнимости формул логики \mathcal{L} имеет разрешающую процедуру с экспоненциальной временной сложностью.

5. Моноадический функциональный символ f будем называть постоянным, а символы g и h — обратимыми, если в каждой интерпретации I для любой формулы p , в I общезначима формула fp или $\neg fp$, соответственно, формула $p \equiv ghp \equiv hgp$.

ТЕОРЕМА 2. Логика \mathcal{L} с постоянными и обратимыми функциональными символами интерпретируется в логике \mathcal{L} .

Д о к а з а т е л ь с т в о. Достаточно определить отображение интерпретации i на внешних атомарных формулах. Представим i в виде композиции, $i\alpha = i_2 i_1 \alpha$, так, что формула $i_1 \alpha$ не имеет вхождения постоянных символов. При построении отображений интерпретации i_1 и i_2 используем схемы с дополнительной магазинной памятью с маркерами [10, 11].

О п р е д е л е н и е. Следующие множества есть типы операторов

- 1) $\{\text{start}\} \times N$ (начальные операторы);
- 2) $N \times \{\text{return } \mathbf{И}, \text{return } \mathbf{Л}\}$ (заключительные операторы);
- 3) $N \times \mathcal{F} \times N$ (вычислительные операторы);
- 4) $N \times (\mathcal{P} \cup \Gamma) \times N \times N$ (условные операторы);
- 5) $N \times \{\text{push}\} \times N_E \times N$ (операторы записи в магазинную память);
- 6) $N \times \{\text{pop}\} \times N^E$ (операторы чтения из магазинной памяти);

где $\text{start}, \dots, \text{pop}$ — специальные символы, Γ — множество формул, $E \geq 1$ — фиксированное натуральное число (E -число схемы) и $N_E = \{1, 2, \dots, E\}$. Натуральные числа в компонентах операторов называются метками.

В модели $t = (I, d)$ каждый оператор s , отличный от начального или заключительного, задает частичное отображение Δ_s на множестве состояний $U \{(N \times D)^n \mid n \geq 1\}$

- $(m, f, m') (m, d, \dots) = (m', f^I(d), \dots)$;
- $(m, p, m_{\mathbf{И}}, m_{\mathbf{Л}}) (m, d, \dots) = (m_{p^I(d)}, d, \dots)$;
- $(m, \text{push}, m', m'') (m, d, \dots) = (m'', d, m', d, \dots)$;

— $(m, \text{pop}, m_1, \dots, m_E) (m, d, m', d', \dots) = (m_{m'}, d', \dots)$;
 — в остальных случаях отображение не определено (точками показаны неизменяемые компоненты состояний.)

Конечное множество операторов называется *схемой* с дополнительной магазинной памятью с маркерами, *размер* схемы есть мощность множества. *Вычисления* схемы \mathcal{Y} в модели t получаются итерацией отношения $\Delta_{(y, t)} = \bigcup \{\Delta_s \mid s \in \mathcal{Y}\}: (m, d_0), \dots$, где m — метка начального оператора из \mathcal{Y} . Если (m', d) — последний элемент вычисления и \mathcal{Y} содержит заключительный оператор $(m', \text{return } \mathbf{I})$ или $(m', \text{return } \mathbf{J})$, то это вычисление имеет, соответственно, значение \mathbf{I} или \mathbf{J} .

В [12] установлена эквивалентность введенного класса схем \mathcal{L} -схемам.

Мы рассмотрим здесь случай АФ1 атомарных формул, случай формул заикливания разбирается аналогично. Транслируем исходную \mathcal{L} -схему α в схему с магазинной памятью R . Мы построим схему R' , эквивалентную R в классе T моделей, в которых постоянные функциональные символы интерпретируются постоянными функциями. С другой стороны, по построению схемы R' в ее вычислениях в любой модели постоянные символы могут применяться только к одному аргументу — входному значению. Это позволяет переопределить модель не изменяя вычислений так, чтобы она принадлежала классу T .

Пусть $C = \{c_1, \dots, c_l\}$ — множество вхождений постоянных символов, а $P = \{p_1, \dots, p_m\}$ — множество меток перехода поп-операторов в R . Предположим для простоты, что схема R детерминированная и постоянные символы не входят в формулы схемы R . Пусть $\Phi \subseteq C^* \times (C \times P)^*$ — (конечное) подмножество списков, не имеющих повторных вхождений элементов C . Каждому списку $\varphi \in \Phi$ поставим в соответствие множество операторов R^φ , которое получается из R последовательным применением следующих операций (первоначально каждой метке i в R соответствует метка i^φ в R^φ).

А. $\varphi = \emptyset$ (пустой список).

1. Оператор (start, i) заменяется на (start, i_1) , где i_1 — новая метка, и добавляется оператор $(i_1, \text{push}, 2E + 1, i)$, где E есть E -число схемы R .

2. Каждый оператор вида (\dots, c_i, j) заменяется на $(\dots, \lambda, j^{(c_i)})$.

Б. $\varphi = ((c_{i_1}, \dots, c_{i_p}), ((c_{j_1}, p_{k_1}), \dots, (c_{j_q}, p_{k_q}))) \neq \emptyset$.

3. Оператор (start, \dots) удаляется.

4. Операторы (\dots, c_{i_1}, \dots) , (\dots, c_{i_p}, \dots) заменяются на $(i_1, \theta), \dots, (i_p, \theta)$ (θ — специальная ф. константа, нигде не определенная при каждой интерпретации).

5. Операторы (\dots, c_i, j) , $c_i \neq c_{i_1}, \dots, c_{i_p}, c_{j_1}, \dots, c_{j_q}$ заменяются на (\dots, λ, j^ψ) , где $\psi = ((c_{i_1}, \dots, c_{i_p}, c_i), ((c_{j_1}, p_{k_1}), \dots, (c_{j_q}, p_{k_q})))$, λ — специальная ф. константа, всегда интерпретируемая как тождественная функция.

Обозначим через Q копию полученного к этому моменту множества операторов (каждой метке i в R^Φ соответствует метка i' в Q). Добавим к R^Φ все операторы из Q .

6. Если $p \geq 1$, то операторы (\dots, c_{j_l}, \dots) , $l = 1, \dots, q$, и (i, pop) заменяются на $(\dots, \lambda, j_{p-1}^\Psi)$, где j_{p-1} — метка перехода оператора $(\dots, c_{i_{p-1}}, j_{p-1})$, если $p > 1$, и j_0 — метка перехода оператора (start, j_0) , и $\Psi = ((c_{i_1}, \dots, c_{i_{p-1}}), ((c_{j_1}, p_{k_1}), \dots, (c_{j_q}, p_{k_q}), (c_{i_p}, p_{k_i})$ (соответственно i))).

7. Операторы (j_1, c_{j_1}, \dots) , (j'_1, c_{j_1}, \dots) , \dots , (j_q, c_{j_q}, \dots) , (j'_q, c_{j_q}, \dots) заменяются, соответственно, на (j_1, λ, k_1) , (j'_1, λ, k_1) , \dots , (j_q, λ, k_q) , (j'_q, λ, k_q) .

8. Оператор (j_i, \dots) , где j_p — метка перехода оператора (\dots, c_{i_p}, j_i) , если $p \geq 1$, и j_0 — метка перехода оператора (start, j_0) , заменяется на (j_p^3, \dots) и добавляются операторы

$$\begin{aligned} &(j_p, \text{pop}, j_p, \dots, j_p \text{ (2E раз)}, j_p^1), \\ &(j_p^1, \text{push}, 2E + 1, j_p^2), \\ &(j_p^2, c_{i_p}, j_p^3), \end{aligned}$$

где j_p^1 , j_p^2 и j_p^3 — новые метки.

9. Операторы (i, push, e, j) заменяются на $(i, \text{push}, e + E, j')$.

10. Операторы $(i', \text{pop}, i'_1, \dots, i'_E)$ (из множества Q) заменяются на $(i', \text{pop}, i'_1, \dots, i'_E, i_1, \dots, i_E, \dots)$.

Построение множества операторов R^Φ закончено. Положим $R' \equiv \bigcup \{R^\Phi \mid \Phi \in \Phi\}$. Очевидно, R' есть схема с E -числом $(2E + 1)$. Заметим, что каждый фрагмент R^Φ , $\Phi \neq \emptyset$, схемы R' состоит из двух «долей». Управление может быть передано во вторую долю только из первой, при выполнении push-оператора (п. 9), и возвращено обратно при чтении записанного маркера (п. 10). Управление может быть также передано из второй доли в другой фрагмент (п. 5). Опишем подробнее процесс моделирования схемы R схемой R' .

Процесс вычисления схемы R' в какой-либо модели (I, d_0) можно рассматривать как «динамическую элиминацию» входящих констант c_1, \dots, c_l . Более точно, R' выполняет вычисление S схемы R , предварительно поместив входное значение d_0 в магазинную память (п. 1), до тех пор пока R не предстоит применить одну из постоянных функций, скажем c_i (если в S не происходит применение констант, то R' заканчивает вычисление или расходуется в соответствии с вычислением R ; в первом случае программа (R', I) имеет то же значение). Тогда управление передается в ту часть схемы R' , метки которой имеют верхний индекс (c_i) (см. пп. 2, 5). Работа каждого такого фрагмента начинается с того, что магазинная память опустошается (что достигается использованием дополнительной информации в E -поле магазинной памяти,

см. п. 1) и в рабочей ячейке восстанавливается входное значение d_0 . К нему применяется функция c_i^I и моделирование продолжается (п. 8) до тех пор пока (если) не произойдет одно из событий:

— R предстоит применить одну из констант c_i в индексе фрагмента схемы R' , в котором находится управление;

— R предстоит применить одну из констант c_j в индексе текущего фрагмента схемы;

— R предстоит применить константу c_j , не входящую в индекс текущего фрагмента R' ;

— R предстоит выполнить рор-оператор при том же состоянии магазинной памяти, которое было при выполнении оператора (\dots, c_i, \dots) .

В соответствии с этим

— схема R' расходится (см. п. 4); заметим, что в этом случае и схема R расходится;

— управление в R' передается в другой фрагмент (см. п. 6, элиминация c_i свелась к элиминации уже элиминированного вхождения константы c_j);

— управление передается в другой фрагмент (п. 5, для элиминации вхождения константы c_j);

— управление передается в другой фрагмент (п. 6, элиминация вхождения константы c_i закончена).

Трансляция $i_1\alpha$ схемы R' в \mathcal{L} -схемы является искомой интерпретацией, поскольку выполнено следующее предложение, доказываемое стандартной индукцией по вычислениям.

Предложение 2. а) $T \models \alpha \equiv i_1\alpha$.

б) Пусть $(I, d_0) \models i_1\alpha$ и интерпретация I' совпадает с I за исключением, быть может, постоянных функциональных символов f , $f'(d) \stackrel{d}{=} f^I(d_0)$. Тогда $(I', d_0) \models \alpha$.

Доказательство теоремы 2 завершается построением интерпретации i_2 аналогично i_1 . Отметим, что в отличие от экспоненциального роста размера схем при отображении i_1 , i_2 увеличивает размер схем линейно. Формула $i\alpha$ не имеет вхождений конструкции заикливания АФ2 или недетерминированных схем, если их не имеет α .

Автор глубоко признателен С. И. Адяну, А. Л. Семенову и М. К. Валиеву за обсуждение работы и ценные замечания.

Московский государственный
университет им. М. В. Ломоносова

Поступило
10.01.89

СПИСОК ЦИТИРОВАННОЙ ЛИТЕРАТУРЫ

- [1] Янов Ю. И. Логические схемы алгоритмов // Проблемы кибернетики. I. 1958. С. 75—127.
- [2] Глушков В. М., Цейтлин Г. Е., Ющенко Е. Л. Алгебры, языки, программирование. Киев: Наукова думка, 1978.
- [3] Семенов А. Л. Некоторые алгоритмические проблемы для Систем Алгоритмических Алгебр // ДАН СССР. 1978. Т. 239, № 5. С. 1063—1066.

- [4] Fisher M., Ladner R. Propositional dynamic logic of regular programs // J. Comput. System Soc. 1979. V. 18. P. 194—211.
- [5] Passy S., Tinchev T. PDL with data constants // Information processing letters. 1985. V. 20, N 1. P. 35—41.
- [6] Valiev M. K. On deterministic PDL with converse operator and constants // Abstracts of 9-th Int. Colloq. on LPMS. Moscow, 17—22 Aug. 1987. V. 1. Moscow, 1987. P. 356.
- [7] Vardi M., Wolper P. Automata theoretic techniques of monadic logics of programs // J. Comput. System. Sci. 1986. V. 32. P. 183—221.
- [8] Rastsvetaev A. L. About recognizability of some properties of monadic schemes with commutative functions // Abstracts of 8-th Int. Colloq. on LPMS. Moscow, 17—22 Aug. 1987. V. 5. Moscow, 1987. P. 161—162.
- [9] Столбоушкин А. П., Тайцлин М. А. Динамические логики // Кибернетика и вычислительная техника. Вып. 2. М., 1986. С. 180—230.
- [10] Tokura N., Kasami T., Furuta Sh. Ianov schemes augmented by a pushdown memory // 15-th Ann. IEEE Symp. on Switching and automata theory. Oct. 1973. P. 84—94.
- [11] Фекличев А. В. Схемы программ с магазином и маркерами и их разрешимые свойства // Изв. АН СССР. Сер. мат. 1984. Т. 5. С. 1060—1077.
- [12] Расцветаев А. Л. Об алгоритмических свойствах некоторых классов монадических схем программ / Деп. в ВИНТИ 31.08.88, № 6804 — В88.