Российская академия наук
Сибирское отделение
Институт систем информатики
им. А. П. Ершова

И. С. Ануреев

СИСТЕМЫ ПЕРЕПИСЫВАНИЯ ФОРМУЛ

Препринт
40

Новосибирск 1997

Предложен новый метод автоматического доказательства, основанный на системах переписывания формул, которые являются обобщением систем переписывания термов. Даны основные понятия теории систем переписывания формул. Рассмотрены свойства корректности и нетеровости для таких систем и сформулированы достаточные условия, гарантирующие выполнение этих свойств.

Siberian Division of the Russian Academy of Sciences
A. P. Ershov Institute of Informatics Systems

I. S. Anureev

FORMULA REWRITING SYSTEMS

Preprint
40

Novosibirsk 1997

A new method of automatic proving is suggested. The method is based on formula rewriting systems, a generalization of term rewriting systems. The main concepts of the theory of formula rewriting systems are given. The properties of correctness and termination for such systems are considered and sufficient conditions which guarantee the fulfilment of these properties are stated.

ВВЕДЕНИЕ

Одним из самых распространенных методов автоматического доказательства является использование систем переписывания термов [1, 2]. Семантика таких систем основана на равенствах, а их применение — на замене равных термов равными. Если применять эти системы к бескванторным формулам, то последние будут переписываться в эквивалентные формулы. Но во многих случаях, например при доказательстве условий корректности, появляющихся при верификации программ, достаточно сохранять при переписывании формул более слабое, чем эквивалентность, свойство истинности формул в некоторой алгебре $\mathcal{A}$, т. е. если формула A переписывается в формулу B, то формула A истинна в алгебре $\mathcal{A}$ тогда и только тогда, когда формула B истинна в алгебре $\mathcal{A}$.

В данной работе предлагается новый метод автоматического доказательства, основанный на так называемых системах переписывания формул — более мощном средстве проведения доказательства в таких областях, в которых требуется сохранять при переписывании формул только свойство истиннности формул.

Кроме сопоставления с образцом и замены сопоставленного терма на новый терм, которые осуществляются при применении обычных систем переписывания термов, эти системы включают также такой разбор случаев и такие замены переменных, которые "упрощают"формулу и в то же время сохраняют истинность формул, т. е. если формула переписывается в конечное множество формул, то она истинна в некоторой алгебре тогда и только тогда, когда все формулы из получающегося конечного множества истинны в этой алгебре.

Такие возможности систем переписывания формул достигаются за счет того, что при переписывании формул с помощью этих систем эквивалентность формул не сохраняется, а сохраняется только истинность формул. Частный случай таких систем — интегрированные системы переписывания — был описан в работе [3].

В отличие от систем переписывания термов, которые предназначены для доказательства равенств посредством сведения левой и правой частей равенства к одному и тому же терму, что обеспечивается такими ключевыми свойствами этих систем, как нетеровость и конфлюентность, гарантирующих единственность нормальной формы, системы переписывания формул предназначены для упрощения произвольных бескванторных формул. Поэтому ключевыми свойствами таких систем

являются свойства нетеровости и корректности в некоторой алгебре (свойство сохранения истинности формул в этой алгебре при их переписывании), гарантирующие, что нормальная форма для любой формулы рано или поздно достигается и истинность исходной формулы определяется истинностью всех формул, входящих в нормальную форму.

В данной работе дается систематическое изложение теории систем переписывания формул, которое включает основные понятия, связанные с такими системами, а также вопросы корректности и нетеровости этих систем.

В п.1 собраны предварительные определения и обозначения, необходимые для последующего изложения. В п.2 определяются основное понятие системы переписывания формул и свойства таких систем. В п.3 формулируется теорема о корректности систем переписывания формул с достаточными условиями, при которых переписывание формул с помощью таких систем сохраняет свойство истинности. В пп.4–6 рассматриваются специальные классы систем переписывания формул, для которых разрешена проблема нетеровости.

1. ПРЕДВАРИТЕЛЬНЫЕ ПОНЯТИЯ

Здесь собраны определения и обозначения, необходимые для последующего изложения.

1.1. Множества и мультимножества

Пусть $|S|$ обозначает мощность множества S, $\mathcal{FS}(S)$ — множество всех конечных подмножеств множества S и $FM(S)$ — множество всех конечных мультимножеств из элементов множества S. Число вхождений элемента u в мультимножество U будем обозначать $occn(u, U)$.

1.2. Термы, формулы, подстановки и контексты

Основные понятия, связанные с системами переписывания термов, читатель может найти в работе [4]. Пусть $\mathcal{F}$ — множество функциональных символов и $\mathcal{X}$ — множество переменных. Тогда $\mathcal{T}(\mathcal{F})$ обозначает множество термов над $\mathcal{F}$, $\mathcal{UF}(\mathcal{F})$ — множество бесквантворных формул с равенством над $\mathcal{F}$, $\mathcal{E}(\mathcal{F}) = \mathcal{T}(\mathcal{F}) \cup \mathcal{UF}(\mathcal{F})$ — множество выражений над $\mathcal{F}$ и $\mathcal{S}(\mathcal{F})$ — множество подстановок над $\mathcal{E}(\mathcal{F})$. Пусть $\mathcal{A}r(f)$ обозначает местность символа $f \in \mathcal{F}$ и $\equiv$ обозначает синтаксическое совпадение выражений.

Будем использовать следующие обозначения для пропозициональных связок и логических констант. Пусть $\Rightarrow$, $\vee$, $\wedge$, $\Leftrightarrow$, $\neg$, T и F — импликация, дизъюнкция, коньюнкция, эквивалентность, отрицание, истинностные значения "истина"и "ложь"соответственно.

Пусть $t \in \mathcal{T}(\mathcal{F})$ и $\sigma \in \mathcal{S}(\mathcal{F})$. Тогда $\mathcal{V}ar(t)$ — множество переменных терма t, $\mathcal{MV}ar(t)$ — мультимножество переменных (с учетом числа вхождений) терма t, $root(t)$ — корень терма t, $size(t)$ — число вхождений функциональных символов в терм t, $\mathcal{D}om(\sigma) = \{x \in \mathcal{X} | x\sigma \not\equiv x\}$ — область определения подстановки σ и $\mathcal{VR}ange(\sigma) = \cup_{x \in \mathcal{D}om(\sigma)} \mathcal{V}ar(x\sigma)$ — область значения подстановки σ.

Пусть $x_1, ..., x_n$ — попарно различные переменные, $t_1, ..., t_n \in \mathcal{T}(\mathcal{F})$. Будем обозначать через $\{x_1 \to t_1, ..., x_n \to t_n\}$ такую подстановку σ, что $\mathcal{D}om(\sigma) \subseteq \{x_1, ..., x_n\}$ и $x_i\sigma \equiv t_i$ для каждого $1 \leq i \leq n$. В частности, ( ) обозначает тождественную подстановку.

Выражение называется линейным, если любая переменная входит в него не более одного раза. Подстановка σ называется линейной на множестве переменных X, если для любых $x, y \in X$ терм $x\sigma$ является линейным, и если $x \not\equiv y$, то термы $x\sigma$ и $y\sigma$ не имеют общих переменных. Контекстами называются выражения из $\mathcal{E}(\mathcal{F} \cup \{\Box\})$, содержащие одно вхождение специального символа $\Box$, который обозначает пустое место. Если $t \in \mathcal{T}(\mathcal{F})$, C — контекст и t подставляется в $\Box$, то результат обозначается $C[t] \in \mathcal{E}(\mathcal{F})$.

1.3. Семантика формул

Напомним некоторые понятия, связанные с семантикой формул. $\mathcal{F}$-алгебра $\mathcal{A}$ задается носителем A и семейством функций $\{f_{\mathcal{A}} : A^n \to A | f \in \mathcal{F}$ и $\mathcal{A}r(f) = n\}$. $ASS_{\mathcal{A}}$ обозначает множество присваиваний $\alpha : \mathcal{X} \to A$, которые расширяются до гомоморфизмов $\mathcal{E}(\mathcal{F}) \to A$. Ограничение присваивания или подстановки α на множество переменных X обозначается $\alpha_{|X}$. Пусть $\mathcal{A}$ — $\mathcal{F}$-алгебра. Общее решение $Sol_{\mathcal{A}}(u)$ формулы u в алгебре $\mathcal{A}$ определяется индуктивно базовыми случаями $Sol_{\mathcal{A}}(T) = \{\alpha \in ASS_{\mathcal{A}}\}$, $Sol_{\mathcal{A}}(F) = \emptyset$, $Sol_{\mathcal{A}}(t = t') = \{\alpha \in ASS_{\mathcal{A}} | \alpha(t) =_{\mathcal{A}} \alpha(t')\}$ и индуктивными шагами $Sol_{\mathcal{A}}(u \wedge u') = Sol_{\mathcal{A}}(u) \cap Sol_{\mathcal{A}}(u')$, $Sol_{\mathcal{A}}(u \vee u') = Sol_{\mathcal{A}}(u) \cup Sol_{\mathcal{A}}(u')$, $Sol_{\mathcal{A}}(\neg u) = ASS_{\mathcal{A}} \setminus Sol_{\mathcal{A}}(u)$, $Sol_{\mathcal{A}}(u \Rightarrow u') = Sol_{\mathcal{A}}(\neg u \vee u')$, $Sol_{\mathcal{A}}(\exists x.u) = \{\alpha \in ASS_{\mathcal{A}} \mid \alpha_{|\mathcal{X} \setminus \{x\}} = \tau_{|\mathcal{X} \setminus \{x\}}$ для некоторого $\tau \in Sol_{\mathcal{A}}(u)\}$ и $Sol_{\mathcal{A}}(\forall x.u) = ASS_{\mathcal{A}} \setminus Sol_{\mathcal{A}}(\exists x.\neg u)$.

Присваивание в $Sol_{\mathcal{A}}(u)$ называется решением формулы u в $\mathcal{A}$. Формула u истинна в $\mathcal{A}$ ($\mathcal{A} \models u$), если любое присваивание является ре-

шением формулы u в $\mathcal{A}$. Истинность в алгебре легко распространяется на конечные мультимножества формул. $U \in \mathcal{FM}(\mathcal{UF}(\mathcal{F}))$ истинно в $\mathcal{A}$ ($\mathcal{A} \models U$), если $\mathcal{A} \models u$ для каждой формулы $u \in U$. Формула u истинна, если $\mathcal{A} \models u$ для каждой $\mathcal{F}$-алгебры $\mathcal{A}$. Две формулы u и u' эквивалентны в $\mathcal{A}$ ($u \equiv_{\mathcal{A}} u'$), если $Sol_{\mathcal{A}}(u) = Sol_{\mathcal{A}}(u)$. Формулы u и u' эквивалентны, если $u \equiv_{\mathcal{A}} u'$ для каждой $\mathcal{F}$-алгебры $\mathcal{A}$.

1.4. Частичные порядки

Напомним основные определения, касающиеся (строгих) частичных порядков. Пусть $X_1, ..., X_n$ — множества, $\succ_1, ..., \succ_n$ — частичные порядки на множествах $X_1, ..., X_n$ соответственно. Тогда $(\succ_1, ..., \succ_n)$ обозначает лексикографическое расширение частичных порядков $\succ_1, ..., \succ_n$ на множество $X_1 \times ... \times X_n$.

Пусть S — множество, $\succ$ — (строгий) частичный порядок на S. Порядок $\succ$ называется нетеровым, если не существует такой бесконечной последовательности $\{s_i\}_{i \in N}$ из элементов множества S, что для любого $i \in N \;\; s_i \succ s_{i+1}$. Бинарное отношение $\succ_m$, определенное на $FM(S)$, называется расширением порядка $\succ$ на мультимножества из $FM(S)$, если для любых мультимножеств $M, M' \in FM(S) \;\; M \succ_m M'$ тогда и только тогда, когда существуют такие мультимножества $X, Y \in FM(S)$, что $\emptyset \neq X \subseteq M$, $M' = (M \setminus X) \cup Y$ и $(\forall y \in Y)(\exists x \in X)\; x \succ y$.

Пусть $\succ$ — (строгий) частичный порядок на $\mathcal{T}(\mathcal{F})$. Он называется

– монотонным, если для любых термов $t, s \in \mathcal{T}(\mathcal{F})$ и контекста C из $t \succ s$ следует $C[t] \succ C[s]$;

– порядком редукции, если $\succ$ является нетеровым и монотонным;

– стабильным относительно подстановок, если для любых термов $t, s \in \mathcal{T}(\mathcal{F})$ и для любой подстановки σ из $t \succ s$ следует $t\sigma \succ s\sigma$;

– термальным порядком, если $\succ$ является стабильным относительно подстановок порядком редукции;

– порядком упрощения, если $\succ$ является монотонным порядком, обладающим свойством подтерма ($C[t] \succ t$ для каждого терма $t \in \mathcal{T}(\mathcal{F})$ и для каждого непустого контекста C).

Пусть $>$ — обычный порядок "больше"на множестве N натуральных чисел с нулем.

1.5. Условные правила переписывания термов и сильные унификаторы

Кроме того, нам потребуются специальный вид условных правил переписывания термов и специальный вид наиболее общих унификаторов подстановок.

Тройка $p|l \to r$ называется условным правилом переписывания термов над $\mathcal{F}$, если $p \in \mathcal{UF}(\mathcal{F}),\ l \in \mathcal{T}(\mathcal{F}) \setminus \mathcal{X},\ r \in \mathcal{T}(\mathcal{F})$ и $\mathcal{V}ar(p) \cup \mathcal{V}ar(r) \subseteq \mathcal{V}ar(l)$. Если посылка $p \equiv T$, она опускается. Конечное множество условных правил переписывания термов над $\mathcal{F}$ называется системой условных правил переписывания термов над $\mathcal{F}$.

Мы ограничимся только такими наиболее общими унификаторами θ термов t и t', что $\mathcal{D}om(\theta) \cup \mathcal{VR}ange(\theta) \subseteq \mathcal{V}ar(t) \cup \mathcal{V}ar(t')$. Аналогичное ограничение принимается для наиболее общего унификатора ϕ подстановок σ и σ': $\mathcal{D}om(\phi) \cup \mathcal{VR}ange(\phi) \subseteq \mathcal{VR}ange(\sigma) \cup \mathcal{VR}ange(\sigma')$.

Пусть $\sigma_1, \sigma_2 \in \mathcal{S}(\mathcal{F})$. Наиболее общий унификатор θ подстановок σ_1 и σ_2 называется сильным унификатором этих подстановок, если формулы $\wedge_{x \in \mathcal{D}om(\sigma_1) \cup \mathcal{D}om(\sigma_2)} x\sigma_1 = x\sigma_2$ и $\wedge_{y \in \mathcal{D}om(\theta)} y = y\theta$ эквивалентны. В этом случае подстановки σ_1 и σ_2 называются сильно унифицируемыми.

Алгоритм нахождения сильного унификатора подстановок σ_1 и σ_2 совпадает с алгоритмом унификации для этих подстановок, если из него удалить правило редукции термов, и задается следующим отношением редукции $\to$ на $FS(UF(F))$:

– $U \to U \setminus \{t = t\}$, если $t = t \in U$;

– $U \to U(x \to t) \cup \{x = t\}$, если $x = t \in U$, $x \notin \mathcal{V}ar(t)$ и x встречается в U ровно один раз;

– $U \to (U \setminus \{t = x\}) \cup \{x = t\}$, если $t = x \in U$ и $t \notin \mathcal{X}$.

Здесь $U \in \mathcal{FS}(\mathcal{UF}(\mathcal{F}))$, $t \in \mathcal{T}(\mathcal{F})$ и $x \in \mathcal{X}$. Алгоритм применяется к множеству формул $W = \{x\sigma_1 = x\sigma_2 | x \in \mathcal{D}om(\sigma_1) \cup \mathcal{D}om(\sigma_2)\}$.

Легко показать, что формулы $\wedge_{u \in U} u$ и $\wedge_{u' \in U'} u'$ эквивалентны, если $U \to U'$. Поэтому, подстановка $\theta = (x_1 \to t_1, ..., x_n \to t_n)$ является сильным унификатором подстановок σ_1 и σ_2, если $\{x_1 = t_1, ..., x_n = t_n\}$ — такая нормальная форма(относительно отношения редукции $\to$) множества формул W, что для всех $1 \le i, j \le n$ $x_i \notin \mathcal{V}ar(t_j)$.

2. СИСТЕМЫ ПЕРЕПИСЫВАНИЯ ФОРМУЛ

Введем основные понятия, связанные с системами переписывания формул.

Определение. *Пусть CR — система условных правил переписывания термов над $\mathcal{F}$ и $s \in \mathcal{T}(\mathcal{F}) \setminus \mathcal{X}$. Пара $r = (CR, s)$ называется правилом переписывания формул над $\mathcal{F}$, если для любого $p|l \to l' \in CR$ термы l и s унифицируемы. Терм s называется образцом r. Конечное множество правил переписывания формул над $\mathcal{F}$ называется системой переписывания формул над $\mathcal{F}$.*

Пример. $(\{f(g(h(y))) \to y,\ f(z) \to z\}, f(g(x))\ )$ — правило переписывания формул, так как образец $f(g(x))$ унифицируем с каждой из левых частей $f(g(h(y)))$ и $f(z)$.

Заметим, что правило переписывания термов $l \to r$ можно рассматривать как правило переписания формул $(\{l \to r\}, l)$. Поэтому системы переписывания термов являются частным случаем систем переписывания формул.

Пусть $r = (\{p_i|l_i \to r_i \mid i \in I\}, s)$ — некоторое правило переписыания формул над $\mathcal{F}$.

Определение. *Семейство подстановок $\{\theta_i^r|i \in I\}$ называется наиболее общим унификатором правила r, если θ_i^r — наиболее общий унификатор термов l_i и s для каждого $i \in I$.*

Пример. Семейство подстановок $\{(x \to h(y)),\ (z \to g(x))\}$ — наиболее общий унификатор правила $(\{f(g(h(y))) \to y,\ f(z) \to z\}, f(g(x))\ )$.

Определение. *Терм t называется редексом правила r, если существует такая подстановка σ^t, что $t \equiv s\sigma^t$, и подстановки σ^t и θ_i^r сильно унифицируемы для каждого $i \in I$. Семейство $\{\phi_i^t|i \in I\}$ сильных унификаторов ϕ_i^t подстановок σ^t и θ_i^r называется заменой переменных, порожденной редексом t. Терм t называется редексом системы переписывания формул R, если t является редексом хотя бы одного правила этой системы.*

Пример. Пусть $r = (\{f(g(x)) \to x\}, f(z))$ — правило переписывания формул. Терм $t \equiv f(y)$ — редекс правила r, так как подстановки $\sigma^t = (z \to y)$ и $\theta^r = (z \to g(x))$ сильно унифицируемы и имеют сильный унификатор $\phi^t = (y \to g(x))$. Тогда $\{\phi^t\}$ — замена переменных, порожденная редексом t. Терм $t \equiv f(g(y))$ не является редексом правила r, так как подстановки $\sigma^t = (z \to g(y))$ и $\theta^r = (z \to g(x))$ не являются сильно унифицируемыми, хотя и являются унифицируемыми.

Определение. *Правило переписывания формул r определяет бинарное отношение редукции $\to_r$ на $\mathcal{FM}(\mathcal{UF}(\mathcal{F}))$ следующим образом: $u \to_r U$, если найдутся контекст C и r-редекс t такие, что $\mathcal{V}ar(C[t]) \cap \mathcal{V}ar(r) = \emptyset$, $u \equiv C[t]$ и $U = \{(p_i\sigma^t \wedge C[r_i\sigma^t])\phi_i^t|i \in I\}$.*

Заметим, что условие $\mathcal{V}ar(C[t]) \cap \mathcal{V}ar(r) = \emptyset$ не является существенным, так как всегда можно переименовать множество переменных правила r. Недостатком этого определения является то, что для правила r вида $\{p_i | s \to r_i \mid i \in I\}, s)$ было бы естественным, чтобы отношение редукции принимало вид $C[t] \to_r \{p_i\sigma^t \wedge C[r_i\sigma^t] | i \in I\}$. Следующий пример показывает, что это не так.

Пример. Для правила $r = (\{f(x) \to x\}, f(x))$ имеем $f(y) = y \to_r x = x$, если в качестве сильного унификатора взять подстановку $(y \to x)$.

Чтобы устранить этот недостаток, будем рассматривать только такие замены переменных, которые состоят из минимальных относительно множества переменных $\mathcal{V}ar(u)$ сильных унификаторов. Введем соответствующее определение.

Определение. *Сильный унификатор ϕ подстановок σ и σ' называется минимальным относительно множества переменных $X \in \mathcal{X}$, если для любого сильного унификатора ϕ' подстановок σ и σ' $|\mathcal{VR}ange(\phi) \cap X| \leq |\mathcal{VR}ange(\phi') \cap X|$.*

Заметим, что алгоритм нахождения сильного унификатора, описанный выше, позволяет для каждого $i \in I$ найти минимальный относительно $\mathcal{V}ar(u)$ сильный унификатор ϕ_i^t подстановок σ^t и ϕ_i^r, если этот алгоритм применяется к множеству формул $\{x\sigma^t = x\phi_i^r | x \in \mathcal{VR}ange(\sigma) \cup \mathcal{VR}ange(\phi_i^r)\}$.

Каждой системе переписывания формул R можно сопоставить абстрактную систему редукций $(\mathcal{FM}(\mathcal{UF}(\mathcal{F})), \{\to_r \mid r \in R\})$. Поэтому, все понятия, определяемые для абстрактных систем редукций (нетеровость, существование нормальной формы и т. д.), можно перенести на системы переписывания формул.

3. КОРРЕКТНОСТЬ СИСТЕМ ПЕРЕПИСЫВАНИЯ ФОРМУЛ

Кроме понятий, перенесенных с абстрактных систем редукций, системы переписывания формул имеют специальное понятие, связанное со спецификой семантики этих систем.

Определение. *Правило r называется корректным в алгебре $\mathcal{A}$, если для любых $U, V \in \mathcal{FM}(\mathcal{UF}(\mathcal{F}))$ из $U \to_r V$ следует, что $\mathcal{A} \models U$ тогда и только тогда, когда $\mathcal{A} \models V$. Система R называется корректной в алгебре $\mathcal{A}$, если все правила этой системы корректны в $\mathcal{A}$.*

Рассмотрим достаточные условия корректности систем переписывания формул. Пусть $u \in \mathcal{UF}(\mathcal{F})$ и $X = \{x_1, ..., x_n\} \subseteq \mathcal{X}$. Будем обозначать

через $\exists X.u$ формулу $\exists x_1. \ldots \exists x_n.u$.

Теорема 1. *Пусть A — $\mathcal{F}$-алгебра и R — система переписывания формул такие, что для любого правила $r = (\{p_i | l_i \to r_i \mid i \in I\}, s) \in R$ следующие формулы истинны в алгебре $\mathcal{A}$:*

– $p_i \Rightarrow l_i = r_i$ для каждого $i \in I$,

– $\exists \mathcal{V}ar(r) \setminus \mathcal{V}ar(s). \vee_{i \in I} (p_i \wedge l_i = s)$,

– $\exists \mathcal{V}ar(r) \setminus \mathcal{V}ar(l_i = s).(p_i \wedge l_i = s \Rightarrow \vee_{j \in I}(p_j \wedge l_j = s \wedge \wedge_{x \in \mathcal{V}ar(r)} x = x\theta_j^r))$ для каждого $i \in I$.

Тогда R корректна в алгебре $\mathcal{A}$.

Доказательство. Пусть $r = (\{p_i | l_i \to r_i \mid i \in I\}, s) \in R$, $u \in \mathcal{UF}(\mathcal{F})$ и $U \in \mathcal{FM}(\mathcal{UF}(\mathcal{F}))$ такие, что $u \to_r U$. Достаточно показать, что $\mathcal{A} \models u$ тогда и только тогда, когда $\mathcal{A} \models U$. Из определения отношения редукции $\to_r$ следует, что существуют контекст C и редекс t правила r такие, что $u \equiv C[t]$, $U = \{(p_i\sigma^t \Rightarrow C[r_i\sigma^t])\phi_i^t | i \in I\}$ и $\mathcal{V}ar(C[t]) \cap \mathcal{V}ar(r) = \emptyset$.

Докажем, что если $\mathcal{A} \models u$, то $\mathcal{A} \models U$. Будем доказывать от противного. Пусть существуют присваивание α и $i \in I$ такие, что $\alpha((p_i\sigma^t \Rightarrow C[r_i\sigma^t])\phi_i^t) = F$. Тогда $\alpha(p_i\sigma^t\phi_i^t) = T$ и $\alpha(C[r_i\sigma^t]\phi_i^t) = F$.

Из первого условия теоремы 1 следует, что $\alpha(l_i\sigma^t\phi_i^t) = \alpha(r_i\sigma^t\phi_i^t)$ и $\alpha(C[l_i\sigma^t]\phi_i^t) = F$. Из определений подстановок θ_i^r и ϕ_i^t имеем $s\theta_i^r \equiv l_i\theta_i^r$ и $x\sigma^t\phi_i^t \equiv x\theta_i^r\phi_i^t$ для каждого $x \in \mathcal{V}ar(r)$. Тогда $s\sigma^t\phi_i^t \equiv s\theta_i^r\phi_i^t \equiv l_i\theta_i^r\phi_i^t \equiv l_i\sigma^t\phi_i^t$. Следовательно, $\alpha(C[s\sigma^t]\phi_i^t) = \alpha(C[l_i\sigma^t]\phi_i^t) = F$. Это противоречит предположению, что $\mathcal{A} \models u$.

Докажем что если $\mathcal{A} \models U$, то $\mathcal{A} \models u$. Будем доказывать от противного. Пусть найдется такое присваивание α, что $\alpha(u) = F$. Из второго условия теоремы следует, что найдутся присваивание α' и $i \in I$ такие, что $\alpha_{|\mathcal{V}ar(u)} = \alpha'_{|\mathcal{V}ar(u)}$, $\alpha'(p_i\sigma^t) = T$ и $\alpha'(l_i\sigma^t) = \alpha'(s\sigma^t)$.

Из третьего условия теоремы следует, что найдутся присваивание α'' и $j \in I$ такие, что $\alpha'' = \alpha'$ всюду, за исключением множества переменных $\mathcal{V}ar(r) \setminus \mathcal{V}ar(l_i = s)$, $\alpha''(p_j\sigma^t) = T$, $\alpha''(x\sigma^t) = \alpha''(x\theta_j^r)$ для каждого $x \in \mathcal{V}ar(r)$ и $\alpha''(l_j\sigma^t) = \alpha''(s\sigma^t)$. Тогда $\alpha''(y) = \alpha''(y\phi_j^t)$ для каждого $y \in \mathcal{V}ar(r) \cup \mathcal{VR}ange(\sigma^t)$, так как ϕ_j^t является сильным унификатором σ^t и θ_j^r.

Следовательно, $\alpha''(p_j\sigma^t\phi_j^t) = \alpha''(p_j\sigma^t) = T$, $\alpha''(C[r_j\sigma^t]\phi_j^t) = F$ и $\alpha''((p_j\sigma^t \Rightarrow C[r_j\sigma^t])\phi_j^t) = F$. Это противоречит предположению, что $\mathcal{A} \models U$. Теорема доказана.

4. УНИФОРМНЫЕ СИСТЕМЫ

Рассмотрим простейший класс систем переписывания формул, для которого пригодны методы доказательства нетеровости, применяемые для обычных систем переписывания термов.

Определение. *Правило* $(\{p_i|l_i \to r_i \mid i \in I\}, s)$ *называется униформным, если* $l_i \equiv s$ *для каждого* $i \in I$. *Система* R *называется униформной, если она состоит только из униформных правил.*

Определение. *Отображение* $Dec : \mathcal{E}(\mathcal{F}) \to \mathcal{FM}(\mathcal{T}(\mathcal{F}))$ *называется декомпозицией выражений, если оно определяется индуктивно базовыми случаями* $Dec(t = t') = \{t, t'\}$, $Dec(T) = \emptyset$ *и* $Dec(F) = \emptyset$ *и индуктивными шагами* $Dec(A * B) = Dec(A) \cup Dec(B)$ *и* $Dec(\neg A) = Dec(A)$, *где* $A, B \in \mathcal{UF}(\mathcal{F})$ *и* $* \in \{\Rightarrow, \vee, \wedge\}$.

Определение. *Система переписывания формул* R *убывает относительно частичного порядка* $\succ$, *если для любого правила* $(\{p_i|l_i \to r_i \mid i \in I\}, s) \in R$ *и для любого* $i \in I$ $\{s\} \succ_m Dec(p_i) \cup \{r_i\}$.

Рассмотрим теперь, как порядки упрощения, применяемые для доказательства нетеровости систем переписывания термов, могут быть использованы для доказательства нетеровости униформных систем.

Теорема 2. *Униформная система является нетеровой, если она убывает относительно некоторого* σ*-стабильного порядка упрощения* $\succ$.

Доказательство. Пусть R — униформная система и $\succ$ — порядок упрощения такие, что R убывает относительно $\succ$. Из нетеровости порядка упрощения $\succ$ следует, что $\succ_m$ и $(\succ_m)_m$ — нетеровые порядки. Поэтому достаточно показать, что

$$\{Dec(u_1), ..., Dec(u_n)\}(\succ_m)_m\{Dec(v_1), ..., Dec(v_m)\},$$

если $\{u_1, ..., u_n\} \to_r \{v_1, ..., v_m\}$ для любого правила $r \in R$.

Из униформности R следует, что для любого правила $r = (\{p_i|l_i \to r_i \mid i \in I\}, s) \in R$, для любого r-редекса t и для любого контекста C отношение редукции $\to_r$ принимает вид $\{C[s\sigma^t]\} \to_r \{p_i\sigma^t \wedge C[r_i\sigma^t] | i \in I\}$. Поэтому достаточно показать, что $Dec(C[s\sigma^t]) \succ_m Dec(p_i\sigma^t \wedge C[r_i\sigma^t])$ для каждого $i \in I$.

Из σ-стабильности $\succ$ и убывания системы R относительно $\succ$ следует, что $s\sigma^t \succ r_i\sigma^t$ и $\{s\sigma^t\} \succ_m Dec(p_i\sigma^t)$. Из монотонности и свойства подтерма порядка упрощения $\succ$ следует, что $Dec(C[s\sigma^t]) \succ_m Dec(p_i\sigma^t) \cup Dec(C[r_i\sigma^t]) = Dec(p_i\sigma^t \wedge C[r_i\sigma^t])$. Теорема доказана.

Заметим, что в этой теореме свойство подтерма существенно для доказательства нетеровости униформных систем. Поэтому для таких систем нет аналога теоремы, которая связывает нетеровость обычных систем переписывания термов с термальными порядками [5].

5. СИСТЕМЫ ЭЛИМИНАЦИИ АНАЛИЗАТОРОВ

Многие системы переписывания формул, встречающиеся на практике, являются конструктивными системами. Система переписывания формул R называется конструктивной, если множество функциональных символов может быть разбито на множество $\mathcal{D}$ определяемых функциональных символов и множество $\mathcal{C}$ конструкторов таких, что для каждого правила $r \in R$ его образец имеет вид $f(t_1, ..., t_n)$ с $f \in \mathcal{D}$ и $t_1, ..., t_n \in \mathcal{T}(\mathcal{C})$. Заметим, что родственное понятие для систем переписывания термов было рассмотрено, например, в [6].

Следующий класс конструктивных систем переписывания формул (системы элиминации анализаторов) позволяет строить упрощающие процедуры, которые элиминируют определяемые функциональные символы (анализаторы). Идея таких систем заключается в просачивании анализаторов через конструкторы вплоть до переменных с последующей их элиминацией с помощью специально подобранных замен переменных. Поэтому такие упрощающие процедуры сводят проверку истинности формул, содержащих анализаторы, к проверке истинности формул без них. К сожалению, даже очень сильные ограничения, накладываемые на системы элиминации анализаторов, гарантируют их терминацию только относительно определенной стратегии редукции.

5.1. Определение систем элиминации анализаторов

Пусть R — конструктивная система переписывания формул над $\mathcal{F}$ с множеством определяемых функциональных символов $\mathcal{D}$ и множеством конструкторов $\mathcal{C}$. Введем сначала некоторые понятия, позволяющие анализировать структуру выражений относительно системы R.

Определение. *Выражение u называется конструктивным, если $u \in \mathcal{E}(\mathcal{C})$. Подстановка σ называется конструктивной, если для любой переменной $x \in \mathcal{X}$ $x\sigma \in \mathcal{T}(\mathcal{C})$.*

Определение. *Выражение u называется вложенным, если найдутся терм t и контексты C_1 и C_2 такие, что $u \equiv C_1[C_2[t]]$, $root(C_2) \in \mathcal{D}$ и $root(t) \in \mathcal{D}$. Выражение u называется простым, если u не явля-*

ется вложенным. Терм t называется вызовом, если $root(t) \in \mathcal{D}$.

Пусть $C_m(u)$ обозначает мультимножество всех простых вызовов, входящих в выражение u.

Определение. *Отображение $\mu_c : \mathcal{E}(\mathcal{F}) \to \mathcal{FM}(N)$ называется мерой конструкторов, если $\mu_c(u) = \{size(t) | t \in C_m(u)\}$. Отображение $Dec_v : \mathcal{E}(\mathcal{F}) \to \mathcal{FM}(\mathcal{X})$ называется декомпозицией переменных, если $Dec_v(u) = \cup_{t \in C_m(u)} \mathcal{MV}ar(t)$.*

Определим теперь системы элиминации анализаторов.

Определение. *Конструктивная FRS R называется системой элиминации анализаторов, если любой простой вызов есть редекс системы R и для любого правила $(\{p_i | l_i \to r_i \mid i \in I\}, s) \in R$ и для любого $i \in I$ выполняются следующие свойства:*

– *θ_i^r — линейная на $\mathcal{V}ar(s)$ и конструктивная подстановка;*

– *p_i и r_i — простые выражения;*

– *p_i и r_i — конструктивные выражения, если $l_i \not\equiv s$;*

– *$Dec_v(s) \supseteq Dec_v(p_i) \cup Dec_v(r_i)$;*

– *$\mu_c(s) >_m \mu_c(p_i)$ и $\mu_c(s) >_m \mu_c(r_i)$.*

Рассмотрим стратегию редукции, гарантирующую терминацию таких систем.

Определение. *Стратегия редукции для систем элиминации анализаторов называется стратегией простого вызова, если правила применяются к таким редексам систем, которые являются простыми вызовами.*

Требование, чтобы любой простой вызов являлся редексом системы R из определения систем элиминации анализаторов, позволяет элиминировать анализаторы. Остальные условия гарантируют терминацию систем элиминации анализаторов относительно стратегии простого вызова.

5.2. Нетеровость систем элиминации анализаторов

Для систем элиминации анализаторов имеет место следующая

Теорема 3. *Любая система элиминации анализаторов является нетеровой относительно стратегии простого вызова.*

Доказательство. Введем сначала несколько дополнительных понятий. Пусть $C_e(u)$ обозначает мультимножество всех вложенных вызовов, входящих в выражение u. Отображение $\mu_a : \mathcal{E}(\mathcal{F}) \to N$ называется мерой анализаторов, если $\mu_a(u) = |C_e(u)|$. Отображение $\mu_v : \mathcal{E}(\mathcal{F}) \to N$

называется мерой переменных, если $\mu_v(u) = \{occn(x, Dec_v(u)) \in N \setminus \{0\} | x \in \mathcal{X}\}$.

Пусть $\succ$ является строгим частичным порядком на $\mathcal{E}(\mathcal{F})$ таким, что $u \succ w$, если $(\mu_a(u), \mu_v(u), \mu_c(u))$ лексикографически больше, чем $(\mu_a(w), \mu_v(w), \mu_c(w))$ с порядками $>$, $>_m$ и $>_m$ на первом, втором и третьем элементах тройки соответственно. Очевидно, что $\succ$ являтся нетеровым.

Поэтому достаточно показать, что для любых $U, V \in \mathcal{FM}(\mathcal{UF}(\mathcal{F}))$ и для любого $r \in R$, если $U \to_r V$ и правило r применяется в соответствии со стратегией простого вызова, то $U \succ_m V$ или, используя свойства расширения порядка на мультимножества, для любого правила $r = (\{p_i | l_i \to r_i \mid i \in I\}, s) \in R$, если $A \equiv C[s\sigma] \to_r \{A_i | i \in I\}$, где $A_i \equiv (p_i\sigma \Rightarrow C[r_i\sigma])\phi_i$ и редекс $s\sigma$ является простым вызовом, то $A \succ A_i$ для каждого $i \in I$.

Согласно определению лексикографического порядка это свойство сводится к следующим свойствам:

1) $\mu_a(A) \geq \mu_a(A_i)$;

2) $\mu_v(A) \geq_m \mu_v(A_i)$, если $\mu_a(A) = \mu_a(A_i)$;

3) $\mu_c(A) >_m \mu_c(A_i)$, если $\mu_a(A) = \mu_a(A_i)$ и $\mu_v(A) = \mu_v(A_i)$.

Рассмотим два случая: $s \equiv l_i$ и $s \not\equiv l_i$.

Пусть $s \equiv l_i$. Тогда $A_i \equiv (p_i\sigma) \wedge C[r_i\sigma]$ и $\mathcal{V}ar(A_i) \subseteq \mathcal{V}ar(A)$. В этом случае доказываемые свойства принимают вид:

1) $\mu_a(A) \geq \mu_a(A_i)$;

2) $Dec_v(A) \supseteq Dec_v(A_i)$, если $\mu_a(A) = \mu_a(A_i)$;

3) $\mu_c(A) >_m \mu_c(A_i)$, если $\mu_a(A) = \mu_a(A_i)$ и $Dec_v(A) \supseteq Dec_v(A_i)$.

Докажем свойство 1. Так как $s\sigma$ — простой вызов, подстановка σ является конструктивной. Согласно определению систем элиминации анализаторов выражения p_i и r_i простые. Тогда выражения $p_i\sigma$ и $r_i\sigma$ также являются простыми, и, следовательно, $\mu_a(p_i\sigma) = 0$ и $\mu_a(r_i\sigma) = 0$. Рассмотрим три случая:

1) не существует таких контекстов C_1 и C_2, что $C \equiv C_1[C_2]$ и C_2 — простой вызов;

2) терм $r_i\sigma$ не является конструктивным, $C \equiv C_1[C_2]$, и C_2 — простой вызов для некоторых контекстов C_1 и C_2;

3) $r_i\sigma$ — конструктивный терм, $C \equiv C_1[C_2]$ и C_2 — простой вызов для некоторых контекстов C_1 и C_2.

В случае 1 $\mu_a(A) = \mu_a(C)$ и $\mu_a(A_i) = \mu_a(C) + \mu_a(r_i\sigma) + \mu_a(p_i\sigma) = \mu_a(C)$. Следовательно, $\mu_a(A) = \mu_a(A_i)$. В случае 2 $\mu_a(A) = \mu_a(C) + 1$

и $\mu_a(A_i) = \mu_a(C) + 1 + \mu_a(r_i\sigma) + \mu_a(p_i\sigma) = \mu_a(C) + 1$. Следовательно, $\mu_a(A) = \mu_a(A_i)$. В случае 3 $\mu_a(A) = \mu_a(C) + 1$ и $\mu_a(A_i) = \mu_a(C) + \mu_a(r_i\sigma) + \mu_a(p_i\sigma) = \mu_a(C)$. Следовательно, $\mu_a(A) > \mu_a(A_i)$.

Докажем свойство 2. Из $\mu_a(A) = \mu_a(A_i)$ следует, что достаточно рассмотреть только первые два случая. В случае 1 $Dec_v(A) = Dec_v(C) \cup Dec_v(s\sigma)$ и $Dec_v(A_i) = Dec_v(C) \cup Dec_v(r_i\sigma) \cup Dec_v(p_i\sigma)$. Из определения систем элиминации анализаторов следует, что $Dec_v(s) \supseteq Dec_v(r_i) \cup Dec_v(p_i)$ и $Dec_v(s\sigma) \supseteq Dec_v(r_i\sigma) \cup Dec_v(p_i\sigma)$. Тогда $Dec_v(A) \supseteq Dec_v(A_i)$. В случае 2 $Dec_v(A) = (Dec_v(C) \setminus Dec_v(C_2)) \cup Dec_v(s\sigma)$ и $Dec_v(A_i) = (Dec_v(C) \setminus Dec_v(C_2)) \cup Dec_v(r_i\sigma) \cup Dec_v(p_i\sigma)$. Из определения систем элиминации анализаторов следует, что $Dec_v(s) \supseteq Dec_v(r_i) \cup Dec_v(p_i)$ и $Dec_v(s\sigma) \supseteq Dec_v(r_i\sigma) \cup Dec_v(p_i\sigma)$. Тогда $Dec_v(A) \supseteq Dec_v(A_i)$.

Докажем свойство 3. Из $\mu_a(A) = \mu_a(A_i)$ следует, что достаточно рассмотреть только два первых случая. В случае 1 $\mu_c(A) = \mu_c(C) \cup \mu_c(s\sigma)$ и $\mu_c(A_i) = \mu_c(C) \cup \mu_c(r_i\sigma) \cup \mu_c(p_i\sigma)$. Из определения систем элиминации анализаторов следует, что $Dec_v(s) \supseteq Dec_v(r_i) \cup Dec_v(p_i)$ и $\mu_c(s) >_m \mu_c(r_i) \cup \mu_c(p_i)$, и, следовательно, $\mu_c(s\sigma) >_m \mu_c(r_i\sigma) \cup \mu_c(p_i\sigma)$ Тогда $\mu_c(A) >_m \mu_c(A_i)$. В случае 2 $\mu_c(A) = (\mu_c(C) \setminus \mu_c(C_2)) \cup \mu_c(s\sigma)$ и $\mu_c(A_i) = (\mu_c(C) \setminus \mu_c(C_2)) \cup \mu_c(r_i\sigma) \cup \mu_c(p_i\sigma)$. Из определения систем элиминации анализаторов следует, что $Dec_v(s) \supseteq Dec_v(r_i) \cup Dec_v(p_i)$ и $\mu_c(s) >_m \mu_c(r_i) \cup \mu_c(p_i)$, и, следовательно, $\mu_c(s\sigma) >_m \mu_c(r_i\sigma) \cup \mu_c(p_i\sigma)$. Тогда $\mu_c(A) >_m \mu_c(A_i)$.

Пусть $s \not\equiv l_i$. Из определения систем элиминации анализаторов следует, что подстановка θ_i является линейной на $var(s)$. Тогда подстановка ϕ_i также является линейной на $var(A)$. В этом случае доказываемые свойства принимают следующий вид:

1) $\mu_a(A) \geq \mu_a(A_i)$;

2) если $\mu_a(A) = \mu_a \cdot A_i$, то для любой переменной $z \in \mathcal{V}ar(A) \setminus \mathcal{D}om(\phi_i)$ $occn(z, Dec_v(A)) \geq occn(z, Dec_v(A_i))$ и для любых переменных $x \in \mathcal{V}ar(A) \cap \mathcal{D}om(\phi_i)$ и $y \in \mathcal{V}ar(x\phi_i)$ $occn(x, Dec_v(A)) > occn(y, Dec_v(A_i))$;

3) если $\mu_a(A) = \mu_a(A_i)$ и $\mathcal{V}ar(s\sigma) = \emptyset$, то $\mu_c(A) >_m \mu_c(A_i)$.

Докажем свойство 1. Согласно определению систем элиминации анализаторов выражения p_i и r_i и подстановка ϕ_i являются конструктивными. Из определения подстановки θ_i и конструктивности подстановок σ и ϕ_i следует, что подстановка θ_i конструктивна. Тогда выражения $p_i\sigma\phi_i$ и $r_i\sigma\phi_i$ конструктивны, и, следовательно, $\mu_a(p_i\sigma) = 0$ и $\mu_a(r_i\sigma) = 0$. Из конструктивности $r_i\sigma\phi_i$ следует, что достаточно рассмотреть два случая:

1) не существует таких контекстов C_1 и C_2, что $C \equiv C_1[C_2]$ и C_2 — простой вызов;

2) $r_i\sigma\phi_i$ — конструктивный терм, $C \equiv C_1[C_2]$ и C_2 — простой вызов для некоторых контекстов C_1 и C_2.

В случае 1 $\mu_a(A) = \mu_a(C)$ и $\mu_a(A_i) = \mu_a(C) + \mu_a(r_i\sigma\phi_i) + \mu_a(p_i\sigma\phi_i) = \mu_a(C)$. Следовательно, $\mu_a(A) = \mu_a(A_i)$. В случае 2 $\mu_a(A) = \mu_a(C) + 1$ и $\mu_a(A_i) = \mu_a(C) + \mu_a(r_i\sigma\phi_i) + \mu_a(p_i\sigma\phi_i) = \mu_a(C)$. Следовательно, $\mu_a(A) > \mu_a(A_i)$.

Докажем свойство 2. Из конструктивности выражений $r_i\sigma\phi_i$ и $p_i\sigma\phi_i$ следует, что $Dec_v(r_i\sigma\phi_i) = \emptyset$ и $Dec_v(p_i\sigma\phi_i) = \emptyset$. Из $\mu_a(A) = \mu_a(A_i)$ имеем, что не существует таких контекстов C_1 и C_2, что $C \equiv C_1[C_2]$ и C_2 — простой вызов. Тогда $Dec_v(A) = Dec_v(C) \cup Dec_v(s\sigma)$ и $Dec_v(A_i) = Dec_v(C\phi_i) \cup Dec_v(r_i\sigma\phi_i) \cup Dec_v(p_i\sigma\phi_i) = Dec_v(C\phi_i)$.

Пусть $x \in \mathcal{V}ar(A) \cap \mathcal{D}om(\phi_i)$ и $y \in \mathcal{V}ar(x\phi_i)$. Докажем, что в этом случае $occn(x, Dec_v(A)) > occn(y, Dec_v(A_i))$ или, после упрощений, $occn(x, Dec_v(C)) + occn(x, Dec_v(s\sigma)) > occn(y, Dec_v(C\phi_i))$. Из линейности ϕ_i имеем $occn(x, Dec_v(C)) = occn(y, Dec_v(C\phi_i))$. Из определения подстановки ϕ_i следует, что $occn(x, Dec_v(s\sigma)) > 0$. Тогда $occn(x, Dec_v(A)) > occn(y, Dec_v(A_i))$.

Пусть $z \in \mathcal{V}ar(A) \setminus \mathcal{D}om(\phi_i)$. Покажем, что тогда $occn(z, Dec_v(A)) \geq occn(z, Dec_v(A_i))$ или, после определенных упрощений, $occn(z, Dec_v(C)) + occn(z, Dec_v(s\sigma)) \geq occn(z, Dec_v(C\phi_i))$. Из $z \notin \mathcal{D}om(\phi_i)$ получим, что $occn(z, Dec_v(C)) = occn(z, Dec_v(C\phi_i))$. Тогда из $occn(z, Dec_v s\sigma)) \geq 0$ следует, что $occn(z, Dec_v(A)) \geq occn(z, Dec_v(A_i))$.

Докажем свойство 3. В этом случае не существует таких контекстов C_1 и C_2, что $C \equiv C_1[C_2]$ и C_2 — простой вызов. Из конструктивности выражений $r_i\sigma\phi_i$ и $p_i\sigma\phi_i$ имеем $\mu_c(r_i\sigma\phi_i) = \emptyset$ и $\mu_c(p_i\sigma\phi_i) = \emptyset$. Тогда $\mu_c(A) = \mu_c(C) \cup \mu_c(s\sigma)$ и $\mu_c(A_i) = \mu_c(C\phi_i) \cup \mu_c(r_i\sigma\phi_i) \cup \mu_c(p_i\sigma\phi_i) = \mu_c(C\phi_i)$. Из $\mathcal{V}ar(s\sigma) = \emptyset$ имеем $\mathcal{V}ar(A) \cap \mathcal{D}om(\phi_i) = \emptyset$ и, следовательно, $\mu_c(C\phi_i) = \mu_c(C)$. Тогда $\mu_c(A) >_m \mu_c(A_i)$, так как $\mu_c(s\sigma) \neq \emptyset$. Теорема доказана.

5.3. Системы элиминации анализаторов и стратегия применения правил

Покажем, что использование стратегии при применении систем элиминации анализаторов является необходимым условием нетеровости таких систем. Пусть система переписывания формул R состоит из следующих правил:

$r1 : (\{f(c(z)) \to z\}, f(x))$,
$r2 : (\{f(c(z)) \to z\}, f(c(z)))$,
$r3 : (\{f(d(x,y)) \to d(f(y), f(x))\}, f(d(x,y)))$,
$r4 : (\{h(c(z)) \to d(h(z), z)\}, h(c(z)))$,
$r5 : (\{h(d(x,y)) \to x\}, h(d(x,y)))$,
$r6 : (\{h(d(x,y)) \to x\}, h(z))$.

Покажем, что система R — система элиминации анализаторов с анализаторами f и h и конструкторами c и d.

Пусть t — простой вызов. Тогда t имеет один из следующих видов: $f(\_)$, $f(c(\_))$, $f(d(\_,\_))$, $h(\_)$, $h(c(\_))$ или $h(d(\_,\_))$. Очевидно, в каждом из этих случаев терм t является редексом системы R. Остальные условия из определения системы элиминации анализаторов принимают для правила r_1 следующий вид:

– подстановка $(x, c(z))$ является конструктивной и линейной на множестве переменных $\{x\}$;

– $f(x)$ и T — простые выражения;

– $f(x)$ и T — конструктивные выражения, если $f(c(z)) \not\equiv f(x)$;

– $\{x\} \supseteq \emptyset$;

– $\{1\} >_m \emptyset$ и $\{1\} >_m \emptyset$.

Для правила r_3 соответствующие условия принимают вид:

– подстановка ( ) является конструктивной и линейной на множестве переменных $\{x, y\}$;

– $f(d(x,y))$ и T — простые выражения;

– $f(d(x,y))$ и T — конструктивные выражения, если $f(d(x,y)) \not\equiv f(d(x,y))$;

– $\{x,y\} \supseteq \{x,y\}$;

– $\{2\} >_m \{1,1\}$ и $\{2\} >_m \{1,1\}$.

Очевидно, что все эти условия выполняются. Аналогичные рассуждения можно провести и для остальных правил системы R. Таким образом, R — система элиминации анализаторов.

Тем не менее следующая цепочка редукций $\underline{d(f(x), f(h(x)))}$ $\to_{r_1} d(z, f(h(c(z)))) \to_{r_4} d(z, f(d(h(z), z))) \to_{r_3} \underline{d(z, \underline{d(f(z), f(h(z)))})} \to_{r_1}$ ... является бесконечной.

Нетеровость системы нарушена, так как правило r_3 было применено к редексу $f(d(h(z), z))$ системы R, который не является простым вызовом.

6. СИСТЕМЫ ЭЛИМИНАЦИИ АНАЛИЗАТОРОВ С АРГУМЕНТНЫМ СТАТУСОМ

Рассмотрим обобщение систем элиминации анализаторов, для которого также имеет место нетеровость относительно стратегии простого вызова.

Пример. Пусть конструктивная система переписывания формул R состоит из единственного правила

$$(\{f(g(g(x)),y) \to f(x,f(a,y))\}, f(g(g(x)),y))$$

с определяемым функциональным символом f и конструкторами g и a. Очевидно, что R является нетеровой относительно стратегии простого вызова. Тем не менее R не является системой элиминации анализаторов, так как терм $f(x,f(a,y))$ является вложенным.

Одним из обобщений систем элиминации анализаторов, позволяющем включать примеры такого рода, является учитывание лишь определенных аргументов функциональных символов при проведении анализа выражений. Так, в этом примере выражение $f(x,f(a,y))$ можно рассматривать как простое, если учитывать только первый аргумент анализатора f. Введем соответствующие определения.

Определение. *Отображение* $arg : \mathcal{D} \to \mathcal{P}(N)$ *называется аргументным статусом, если* $arg(f) \subseteq \{1,...,\mathcal{A}r(f)\}$.

Определения вложенного выражения, меры констукторов и меры переменных изменяются с учетом аргументного статуса.

Определение. *Выражение* u *назыветсяся вложенным относительно* arg, *если найдутся контекст* C *и терм* $f(t_1,...,t_n)$ *такие, что* $u \equiv C[f(t_1,...,t_n)]$, $f \in \mathcal{D}$ *и терм* t_i *не является конструктивным для некоторого* $i \in arg(f)$. *Выражение* u *называется простым относительно* arg, *если* u *не является вложенным относительно* arg.

Пусть $C_m(u)$ обозначает мультимножество всех простых относительно arg вызовов, входящих в выражение u, $Imm(t)$, где $t \equiv f(t_1,...,t_n) \in \mathcal{T}(\mathcal{F})$, обозначает мультимножество $\{t_i | i \in arg(f)\}$.

Определение. *Отображение* $\mu_c : \mathcal{E}(\mathcal{F}) \to \mathcal{FM}(N)$ *называется мерой конструкторов, если* $\mu_c(u) = \{\sum_{t' \in Imm(t)} size(t') | t \in C_m(u)\}$. *Отображение* $Dec_v : \mathcal{E}(\mathcal{F}) \to \mathcal{FM}(\mathcal{X})$ *называется декомпозицией переменных, если* $Dec_v(u) = \cup_{t \in C_m(u)} \cup_{t' \in Imm(t)} \mathcal{MV}ar(t')$.

Определим теперь системы элиминации анализаторов с аргументным статусом.

Определение. *Конструктивная система переписывания формул R называется системой элиминации анализаторов с аргументным статусом arg, если любой простой вызов есть редекс системы R и для любого правила $(\{p_i|l_i \to r_i \mid i \in I\}, s) \in R$ и для любого $i \in I$ выполняются следующие свойства:*

– *θ_i^r — линейная на $\mathcal{V}ar(s)$ и конструктивная подстановка;*

– *p_i и r_i — простые относительно arg выражения;*

– *p_i и r_i — конструктивные выражения, если $l_i \not\equiv s$;*

– *$Dec_v(s) \supseteq Dec_v(p_i) \cup Dec_v(r_i)$;*

– *$\mu_c(s) >_m \mu_c(p_i)$ и $\mu_c(s) >_m \mu_c(r_i)$;*

– *$\mathcal{V}ar(s) \cap \mathcal{D}om(\theta_i^r) \subseteq \cup_{s' \in Imm(s)} \mathcal{V}ar(s')$.*

Первые пять условий аналогичны соответствующим условиям из определения системы элиминации анализаторов. Дополнительное шестое условие гарантирует, что все заменяемые переменные учитываются в образце s и, таким образом, в редексах.

Нетрудно показать, что системы элиминации анализаторов с аргументым статусом также являются нетеровыми относительно стратегии простого вызова.

Теорема 4. *Любая система элиминации анализаторов со статусом аргументов является нетеровой относительно стратегии простого вызова.*

Доказательство этой теоремы аналогично доказательству соответствующей теоремы для систем элиминации анализаторов.

7. ЗАКЛЮЧЕНИЕ

В данной работе не рассматривался такой важный вопрос, как использование систем переписывания формул в качестве средства разработки процедур автоматического доказательства.

Конкретные процедуры автоматического доказательства, основанные на системах переписывания формул, и их применение в проблемно-ориентированной верификации (в частности, для доказательства условий корректности для простых программ сортировки массивов и как средства элиминации таких структур данных, как массивы, последовательные файлы и списки) можно найти в работе [3].

Построение семейства процедур автоматического доказательства на основе правил переписывания формул — важная часть блока доказательства проблемно-ориентированной системы верификации СПЕКТР-2 [7], разрабатываемой в лаборатории теоретического программирова-

ния ИСИ СО РАН. В частности, применение систем переписывания формул позволило автоматически доказать корректность программы сортировки файлов естественным слиянием.

СПИСОК ЛИТЕРАТУРЫ

1. **Hsiang J.** Refutational theorem proving using term-rewriting systems // Artif. Intell. — 1985. — Vol. 25. — P. 255–300.
2. **Hsiang J., Kirchner H., Lescanne P., Rusinowitch M.** Automated theorem proving in the presence of equalities // J. Automat. Reasech. — 1992. — Vol. 14. — P. 71–100.
3. **Анореев И.С.** Интегрированные правила переписывания термов и их применение в автоматической верификации программ // Проблемы спецификации и верификации параллельных систем. — Новосибирск, 1995. — С. 185–213.
4. **Dershowitz N., Jouannaud J.-P.** Rewrite systems // Handbook of Theoretical Computer Science. — 1990. — Vol. 6. — P. 243–320.
5. **Steinbach J.** Simplification ordering: history of results // Fundamenta Informaticae. — 1995. — Vol. 24, № 1. — P. 47–87.
6. **Klop J.-W.** Term rewriting systems // Handbook of Logic in Computer Science. — 1993. — Vol. 2. — P. 1–116.
7. **Nepomniaschy V.A., Sulimov A.A.** Problem-oriented means of program specification and verification in project SPECTRUM // Lect. Notes Comput. Sci. — 1993. — Vol. 722 — P. 374–378.

И. С. Ануреев

СИСТЕМЫ ПЕРЕПИСЫВАНИЯ ФОРМУЛ

Препринт
40

Рукопись поступила в редакцию 20.05.97
Рецензент Шилов Н. В.
Редактор Л. А. Карева

Подписано в печать 29.05.97
Формат бумаги 60×84 1/16 Объем 1,3 уч.-изд.л., 1,4 п.л.
Тираж 100 экз.
