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

И. С. Ануреев

ТЕОРИЯ СИСТЕМ ПЕРЕПИСЫВАНИЯ ФОРМУЛ

Препринт
54

Новосибирск 1998

Представлено систематическое изложение теории систем переписывания формул. Предложен унифицированный подход к определению корректности таких систем. Рассмотрена связь систем переписывания формул с теориями сортов. Дан обзор классов завершимых систем переписывания формул.

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

I. S. Anureev

FORMULA REWRITING SYSTEM THEORY

Preprint
54

Novosibirsk 1998

Formula rewriting system theory is represented. The unified approach to correctness definition of such systems is suggested. A relation between formula rewriting systems and type theories is considered. A survey of classes of terminating formula rewriting systems is given.

ВВЕДЕНИЕ

Широко распространенным средством проблемно-ориентированного подхода к автоматическому доказательству теорем являются техники переписывания формул. Среди них можно выделить технику переписывания термов [9, 10, 35, 15], основанную на системах переписывания термов (СПТ), и технику сужения [24].

Техника переписывания термов [13, 29–32, 42] позволяет приспосабливать доказательство к конкретным проблемным областям, для чего достаточно лишь модифицировать СПТ, лежащую в основе разрешающей процедуры. СПТ довольно эффективны, так как основаны на простом механизме сопоставления с образцом и не требуют разбора случаев. В то же время весьма часто встречаются формулы, не позволяющие проводить доказательства по типу замены равных равными, лежащему в основе СПТ. К тому же при проведении доказательства часто требуется разбор случаев. И хотя были разработаны различные обобщения СПТ:

– условные СПТ [16, 20, 38, 45, 50, 56], допускающие применение правила переписывания при выполнении определенного условия;

– эквациональные СПТ [14, 34, 37, 49, 55], комбинирующие применение правил переписывания с выводом в эквациональных теориях;

– СПТ со встроенными разрешающими процедурами [4–7];

– СПТ для теорий, которые не являются квазиэквациональными, в частности для логики первого порядка [40, 48];

– СПТ с разбором случаев [8],

они во многих случаях оказались недостаточными для проведения полностью автоматического доказательства, в частности разбор случаев в них не достаточно мощен, нигде не используется такая упрощающая техника, как замена переменных.

Техника сужения представляет собой объединение переписывания и унификации. Эта техника проверки выполнимости формул исторически возникла как средство решения проблемы унификации в эквациональных теориях (E-унификации), была распространена на условные равенства [17, 18, 33, 39] и нашла применение как механизм, позволяющий объединять логические и функциональные языки [25–28, 44]. Разрешающие процедуры, основанные на сужении, также легко модифицируются при переходе от одной проблемной области к другой, как и процедуры, основанные на переписывании. Такая техника позволяет находить конкретные символические решения (унификаторы) и даже множество всех основных символических решений (наиболее общих унификаторов), если система переписывания, лежащая в их основе, является полной (нетеровой и конфлюентной). Унификация, используемая при сужении, позволяет моделировать такое мощное упрощение формул, как замена переменных. В то же время перебор, возникающий при применении техники сужения, создает большое пространство поиска, что приводит к быстрому исчерпанию скоростных и временных ресурсов машины. Еще одной сложностью, связанной с сужением, является более сложный механизм применения этой техники по сравнению с СПТ, требующий вычисления при каждом применении правила переписывания наиболее общего унификатора. Для решения этой проблемы разработано множество стратегий [21, 22] (нормальное сужение [23], базисное сужение [36], комбинация базисного и нормального сужений [46, 51],стратегия "самый внутренний"[25], ленивое сужение [54]), позволяющих уменьшить перебор, но эти стратегии, не привязанные к конкретным проблемным областям, лишь незначительно уменьшают перебор. Другое направление, в котором ведутся исследования, — установление полноты сужения при более слабых ограничениях, накладываемых на СПТ. Здесь получены некоторые результаты для конфлюентных [36] и полуполных (semi-complete) [53] СПТ. К сожалению, ориентированность техники сужения в основном на разрешение равенств в эквациональных теориях, а также несохранение даже выполнимости, не говоря уже об эквивалентности, при таком преобразовании формул привело к тому, что эта перспективная техника не используется в достаточной степени в автоматическом доказательстве теорем.

Из вышесказанного следует, что создание новой техники автоматического доказательства, которая бы сочетала достоинства вышеизложенных методов и техник и была свободна от их недостатков, является актуальной задачей в области автоматического доказательства теорем. Теория систем переписывания формул (СПФ), предложенная автором в работах [1, 2], является попыткой решения этой задачи.

Техника переписывания, основанная на СПФ, наследует достоинства техник переписывания и сужения и лишена большинства из перечисленных недостатков. Система переписывания формул состоит из конечного множества правил переписывания формул, каждое из которых задается некоторой условной системой переписывания термов — основой правила и некоторым выражением — образцом правила. Если ограничиться только применением СПФ к равенствам, то применение правила переписывания формул к некоторому равенству можно представить как одновременное сужение этого равенства в одной и той же позиции при помощи каждого правила из основы. В результате получим ровно столько "суженных"равенств, сколько правил входит в основу, моделируя таким образом разбор случаев. Преимущество техники переписывания формул, основанной на СПФ, по сравнению с техникой сужения проявляется в том, что позиция в равенстве, к которой применимо правило переписывания формул, вычисляется проще, чем для сужения. Для этого необходимо выполнить только сопоставление с образцом и алгоритм синтаксической унификации без правила редукции термов. Пространство поиска при применении СПФ также сокращается по сравнению с сужением, так как применяются не произвольные правила из некоторой СПТ, как в технике сужения, а только конкретные выбранные подмножества этой СПТ — основы правил, входящих в эту СПФ. Но самое главное преимущество СПФ в их применении к произвольным замкнутым $\forall$- и $\exists$-формулам, а не только к равенствам. Более того, в качестве теорий, в рамках которых проводится доказательство, можно использовать теории с кванторами, а не только эквациональные или квазиэквациональные (с условными равенствами) теории. Это происходит за счет того, что условия корректности для правил переписывания формул кроме равенств и условных равенств включают также формулы вида

$$\forall x_1 ... \forall x_n \exists y_1 ... \exists y_m A,$$

где A — произвольная бескванторная формула. Само понятие корректности для правил переписывания формул также изменяется. Если для правил переписывания термов оно обычно означает сохранение эквивалентности бескванторных формул при переписывании относительно некоторой алгебры, то для правил переписывания формул понятие корректности означает сохранение эквивалентности замкнутых $\forall$- и $\exists$-формул относительно некоторой алгебраической системы.

СПФ, соединяющие в себе элементы техники переписывания и техники сужения, хорошо сочетаются также с универсальными методами автоматического доказательства и разрешающими процедурами для конкретных теорий. Так, связь с универсальными методами проявляется на уровне доказательства условий корректности для СПФ. Эти условия образуют достаточно широкий класс кванторных формул, доказательство которых лучше всего проводить с помощью универсальных методов. СПФ предназначены прежде всего для использования в качестве упрощающих процедур, сохраняющих выполнимость (истинность) формул. Тогда, чтобы получить на их основе процедуры проверки выполнимости (истинности) формул некоторых теорий, требуется комбинирование СПФ с разрешающими процедурами для теорий, формулы которых являются нормальными формами относительно применяемых СПФ. Использование СПФ в качестве упрощающих процедур ставит проблему выделения классов систем переписывания формул, которые являлись бы нетеровыми или хотя бы завершались относительно некоторой заданной стратегии применения правил. Ряд таких классов был выделен в наших работах [1, 2, 12].

Данная работа представляет собой систематическое изложение теории СПФ. Отметим только некоторые новые моменты. Если в ранних работах делался акцент либо на сохранение выполнимости [3, 12], либо на сохранение истинности [1, 2] при переписываниях формул, то здесь принят унифицированный подход к определению корректности таких систем. СПФ переписывают замкнутые $\forall$- и $\exists$-формулы в эквивалентные формулы. Более точно, с учетом разбора случаев конъюнкция замкнутых $\forall$-формул и дизъюнкция замкнутых $\exists$-формул переписываются в эквивалентные формулы. Рассмотрена связь корректности систем переписывания формул с теориями сортов.

В работе наиболее полно представлены вопросы терминации СПФ. Дается доказательство неразрешимости проблемы нетеровости для СПФ. Изменено определение систем элиминации анализаторов и систем элиминации анализаторов со статусом, завершающихся относительно стратегии "любой самый внутренний", что позволило расширить эти классы систем. Добавлен новой класс систем, завершающихся относительно этой стратегии, — системы элиминации анализаторов с базой подстановок.

В п.1 собраны логические и теоретико-множественные обозначения, а также необходимые сведения из теории бинарных отношений и частичных порядков. В п.2 определяются основное понятие системы переписывания формул и свойства таких систем. В п.3 описывается семантика СПФ и формулируется теорема о корректности систем переписывания формул. В п.4 выделяется специальный класс теорий сортов — теории сорта, определяемые обобщенными конструкторами, и исследуется связь этого класса с корректностью СПФ. В пп.5—8 рассматриваются специальные классы систем переписывания формул, для которых даются методы доказательства завершимости. В п.5 описывается простейший класс СПФ — униформные системы, для которого показывается неразрешимость проблемы нетеровости, что приводит к неразрешимости этой проблемы и для класса всех СПФ. Рассматривается связь нетеровости униформных систем с порядками упрощения. Остальные пункты посвящены специальным классам конструктивных систем — систем, в которых множество функциональных и предикатных символов разбито на два непересекающихся класса — конструкторы и анализаторы. В пп.6—8 описываются классы конструктивных систем, позволяющих элиминировать анализаторы, а также даются модифицированные определения и основные свойства систем элиминации анализаторов и систем элиминации анализаторов со статусом соответственно. В п.8 вводится новый класс конструктивных систем — системы элиминации анализиторов с базой подстановок, которые, как и предыдущие два класса систем, завершаются относительно стратегии "любой самый внутрений". Основные результаты и возможные направления дальнейшего развития аппарата СПФ приводятся в заключении.

Работа поддержана частично грантом ИНТАС-РФФИ 95-0378. Автор благодарен Валерию Александровичу Непомнящему за конструктивную критику и полезные замечания.

1. ПРЕДВАРИТЕЛЬНЫЕ СВЕДЕНИЯ

1.1. Логические и теоретико-множественные обозначения

Считая, что читатель знаком с основными понятиями логики первого порядка и теорией унификации, ниже приводим используемую нотацию.

Пусть S — множество, M — мультимножество, $m \in M$. Тогда $|S|$ обозначает мощность S; $\mathcal{FS}(\mathcal{S})$ — множество всех конечных подмножеств S; $\mathcal{FM}(\mathcal{S})$ — множество всех конечных мультимножеств, состоящих из элементов S; и $\mathcal{O}(m, M)$ — число вхождений элемента m в M. Пусть $\Sigma = (\mathcal{F}, \mathcal{P}, \mathcal{V})$ — сигнатура с множеством функциональных символов $\mathcal{F}$, множеством предикатных символов $\mathcal{P}$ и множеством переменных $\mathcal{V}$. Тогда $\mathcal{T}$ обозначает множество всех Σ-термов; $\mathcal{UF}$ — множество всех бескванторных Σ-формул с равенством $=$; $\mathcal{E} = \mathcal{T} \cup \mathcal{UF}$ — множество всех Σ-выражений и $\mathcal{S}$ — множество всех подстановок над $\mathcal{E}$.

Пусть $u \in \mathcal{E}$, $E \subseteq \mathcal{E}$. Обозначим $\mathcal{V}ar(u)$, $\mathcal{MV}ar(u)$ соответственно множество и мультимножество (с учетом числа вхождений) переменных, входящих в u; $\mathcal{MV}ar_E(u)$ — мультимножество переменных, входящих в u, за исключением переменных, входящих в подвыражения u, принадлежащие E; $root(u)$ — корень u; $|u|$ — число переменных, функциональных и предикатных символов, входящих в u; $|u|_E$ — число переменных, функциональных и предикатных символов, входящих в u, за исключением тех, которые входят в подвыражения u, принадлежащие E; $\mathcal{P}(u)$ — множество позиций в u с Λ как самой верхней позицией; $\mathcal{D}om(\sigma)$, $\mathcal{VR}ange(\sigma)$ — область определения и область значения σ соответственно.

Пусть $x_1, ..., x_n$ — попарно различные переменные, $t_1, ..., t_n \in \mathcal{T}$. Обозначим

$$(x_1 \to t_1, ..., x_n \to t_n)$$

подстановку σ такую, что $\mathcal{D}om(\sigma) \subseteq \{x_1, ..., x_n\}$ и $x_i\sigma = t_i$ для каждого i. Тождественная подстановка обозначается ( ).

1.2. Бинарные отношения и частичные порядки

Приведем некоторые сведения о нетеровых бинарных отношениях и частичных порядках. Подробное рассмотрение различных частичных порядков может быть найдено в [52].

Определение 1.1. Пусть S — множество, $\to$ — бинарное отношение на S, $\succ$ — строгий частичный порядок на S. Отношение $\to$ называется нетеровым, если не существует бесконечной цепи вида

$$s_0 \to s_1 \to \ldots .$$

Бинарное отношение $\succ_m$, определенное на $\mathcal{FM}(S)$, называется расширением порядка $\succ$ на мультимножества из $\mathcal{FM}(S)$, если для любых мультимножеств $M, M' \in \mathcal{FM}(S)$ $M \succ_m M'$ 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 для любого $y \in Y$ найдется $x \in X$ такой, что $x \succ y$. Пусть $S_1, ..., S_k$ — множества с частичными порядками $\succ_1, \ldots \succ_k$ соответственно, $s_i, s'_i \in S_i$ для каждого $1 \leq i \leq k$. Будем говорить, что кортеж $(s_1, ..., s_k)$ лексикографически больше кортежа $(s'_1, ..., s'_k)$, если $s_1 \succ_1 s'_1$ или $s_1 = s'_1$ и $(s_2, ..., s_k)$ лексикографически больше кортежа $(s'_2, ..., s'_k)$; (s_k) лексикографически больше (s'_k), если $s_k \succ_1 s'_k$. Получающийся частичный порядок называется лексикографичским порядком, порожденным порядками $\succ_1, \ldots \succ_k$.

Известно, что если $\succ, \succ_1, ..., \succ_k$ — нетеровы порядки, то порядки $\succ_m$ и лексикографический порядок, порожденный порядками $\succ_1, ... \succ_k$, также являются нетеровыми.

Основным свойством рассматриваемых в этой работе систем переписывания формул (СПФ), так же как и систем переписывания термов (СПТ), является свойство завершимости (нетеровости). Следующие порядки, применяемые при доказательстве нетеровости СПТ, использованы ниже при доказательстве завершимости СПФ.

Определение 1.2. Пусть $\succ$ — строгий частичный порядок на $\mathcal{T}$, $t, s \in \mathcal{T}$, $f \in \mathcal{F}$, $\sigma \in \mathcal{S}$. Будем говорить, что $\succ$

– монотонный, если $s \succ t \Rightarrow f(..., s, ...) \succ f(..., t, ...)$ для любых s, t, f;

– порядок редукции, если $\succ$ монотонен и нетеров;

– стабильный относительно подстановок, если $t \succ s \Rightarrow t\sigma \succ s\sigma$ для любых s, t, σ;

– термальный порядок, если $\succ$ — стабильный относительно подстановок порядок редукции;

– обладает свойством подтерма, если $s \succ t \Rightarrow f(..., t, ...) \succ t$ для любых t, f;

– порядок упрощения, если $\succ$ монотонен и обладает свойством подтерма.

В дальнейшем множество натуральных чисел обозначается $\mathcal{N}$ и $>$ обозначает естественное упорядочивание на $\mathcal{N}$.

2. СИСТЕМЫ ПЕРЕПИСЫВАНИЯ ФОРМУЛ

В этом пункте рассмотрены основные синтаксические конструкции, связанные с системами переписывания формул. Семантика СПФ, основанная на понятии корректности СПФ, дается ниже.

2.1. Определение

Системы переписывания формул определяются следующим образом.

Определение 2.1. Пусть $\mathcal{B} \in FS(\mathcal{UF} \times \mathcal{E} \times \mathcal{E})$, $s \in \mathcal{E}$. Пара $\rho = (\mathcal{B}, s)$ называется правилом переписывания формул, если для любой тройки $(p, l, r) \in \mathcal{B}$ выражения l и s унифицируемы; $\mathcal{B}$ называется основой ρ, (p, l, r) — ветвью ρ, s — образцом ρ. Множество правил переписывания формул образует систему переписывания формул (СПФ). В дальнейшем будем рассматривать только конечные системы переписывания формул и писать $p|l \to r$ вместо (p, l, r). Пусть $\pi = p|l \to r$ — ветвь правила ρ. Будем называть p посылкой, l — левой частью и p — правой частью ветви π. В случае, когда ветвь имеет вид $true|l \to r$, посылка $true$ будет опускаться.

Пример 2.2. Пусть

$$\mathcal{B} = \{h(f(x)) \to x, h(g(x)) \to x\}.$$

Тогда $\rho = (\mathcal{B}, h(y))$ — правило переписывания формул с основой $\mathcal{B}$, состоящей из двух ветвей $h(f(x)) \to x$ и $h(g(x)) \to x$ и образцом $h(y)$, так как образец $h(y)$ унифицируем как с левой частью $h(f(x))$ первой ветви, так и с левой частью $h(g(x))$ второй ветви основы $\mathcal{B}$.

Прежде чем определять отношение переписывания (редукции), порождаемое СПФ, введем понятие тривиального унификатора — специального случая наиболее общего унификатора (НОУ).

Определение 2.3. Пусть

$$\bar{x} = \mathcal{D}om(\sigma) \cup \mathcal{D}om(\sigma'),\ \bar{y} = \mathcal{V}ar(\bar{x}\sigma = \bar{x}\sigma'),\ \bar{z} = \mathcal{VR}ange(\theta).$$

НОУ θ подстановок σ и σ' называется тривиальным унификатором σ и σ', если формула

$$\forall\bar{y}\exists\bar{z}(\bar{x}\sigma = \bar{x}\sigma' \Rightarrow \bar{y} = \bar{y}\theta)$$

тождественно-истинна. Подстановки σ и σ' тривиально унифицируемы, если существует тривиальный унификатор этих подстановок.

Алгоритм поиска тривиального унификатора представляет собой вырожденный случай алгоритма унификации [43] и получается из него удалением правила редукции термов.

Пример 2.4. Пусть

$$\sigma = (x \to f(y)),\ \sigma' = (x \to z, y \to f(z)).$$

Покажем, что σ и σ' тривиально унифицируемы и

$$\theta = (y \to f(z'), z \to z')$$

— тривиальный унификатор σ и σ'. Действительно, так как θ — НОУ σ и σ', достаточно доказать, что формула

$$\forall y\forall z\exists z'(f(y) = z \wedge y = f(z) \Rightarrow y = f(z') \wedge z = z')$$

тождественно-истинна и переписывается в эквивалентную формулу

$$\forall y \forall z (f(y) = z \wedge y = f(z) \Rightarrow y = f(z)),$$

тождественная истинность которой очевидна.

Пример 2.5. Пусть

$$\sigma = (x \to f(y)),\ \sigma' = (x \to f(z)).$$

Тогда σ и σ' унифицируемы и

$$\theta = (y \to z', z \to z')$$

— НОУ σ и σ'. Тем не менее, σ и σ' не тривиально унифицируемы, так как формула

$$\forall y \forall z \exists z' (f(y) = f(z) \Rightarrow y = z' \wedge z = z')$$

не тождественно-истинна. Действительно, она переписывается в эквивалентную формулу

$$\forall y \forall z (f(y) = f(z) \Rightarrow y = z),$$

ложную на классе моделей, определяемом аксиомами

$$\neg(a = b) \text{ и } f(a) = f(b),$$

где a, b — некоторые константы.

2.2. Отношение редукции

Отношение переписывания для СПФ строится таким образом, чтобы при выполнении определенных условий замкнутые $\forall$- и $\exists$-формулы переписываются в эквивалентные формулы. И те и другие формулы можно представлять как бескванторные формулы со свободными переменными, но с различной семантикой, определяемой способом связывания свободных переменных (только квантором всеобщности или только квантором существования). Все используемые в этом и следующих пунктах семантические понятия определяются относительно некоторой фиксированной алгебраической системы $\mathcal{A}$, которая опускается. Так, например, под истинностью формулы A подразумевается истинность A в алгебраической системе $\mathcal{A}$.

Определение 2.6. Пусть $A \in \mathcal{UF}$, $\bar{x} = \mathcal{V}ar(A)$. Будем говорить, что формула A экзистенциально связана, если истинность (семантика) A определяется как истинность формулы $\exists\bar{x}(A)$. Аналогично, формула A универсально связана, если истинность A определяется как истинность формулы $\forall\bar{x}(A)$.

Особенностью СПФ является тот факт, что они переписывают формулу в конечное мультимножество формул, а не просто в формулу, в конечном счете переписывая мультимножества формул в мультимножества формул. Расширим введенное определение на конечные мультимножества бескванторных формул со свободными переменными.

Определение 2.7. Пусть $U \in \mathcal{FM}(\mathcal{UF})$, $A \in U$, $\bar{x}_A = \mathcal{V}ar(A)$. Будем говорить, что U экзистенциально связано, если истинность U определяется как истинность формулы $\vee_{A\in U}\exists\bar{x}_A(A)$. Аналогично, U универсально связано, если истинность U определяется как истинность формулы $\wedge_{A\in U}\forall\bar{x}_A(A)$.

В дальнейшем считаем, что каждой формуле и мультимножеству формул приписан тип связывания. Определим отношение переписывания (редукции), порождаемое СПФ.

Определение 2.8. Пусть $\rho = (\mathcal{B}, s) \in R$, $\pi = (p|l \to r)$ — ветвь ρ, θ_π — НОУ l и s, $A \in \mathcal{FM}(\mathcal{UF})$, $q \in \mathcal{P}(A)$, $\sigma \in \mathcal{S}$, $s\sigma = A_q$, σ и θ_π тривиально унифицируемы, ϕ_π — тривиальный унификатор σ и θ_π. Чтобы избежать конфликта переменных, будем считать, что A, R, $\mathcal{VR}ange(\theta_\pi)$, $\mathcal{VR}ange(\phi_\pi)$ не имеют общих переменных,

$$\begin{array}{l}\mathcal{D}om(\theta_\pi) = \mathcal{V}ar(l) \cup \mathcal{V}ar(s),\\ \mathcal{D}om(\phi_\pi) = \mathcal{V}ar(\{x\sigma = x\theta_\pi | x \in \mathcal{V}ar(l) \cup \mathcal{V}ar(s)\}),\\ \mathcal{D}om(\sigma) = \mathcal{V}ar(s).\end{array}$$

Отношение редукции $\to_R$, порождаемое R, сохраняет тип связывания формул и мультимножеств формул и определяется следующим образом:

$$\to_R = \{(A, \{(p\sigma * A[r\sigma]_q)\phi_\pi | \pi\}) | \text{по всем } A, q, \rho\},$$

где $*$ имеет вид $\wedge$ при экзистенциальном связывании и $\Rightarrow$ при универсальном связывании. Для $p = true$ формула $p\sigma * A[r\sigma]_q$ принимает вид $A[r\sigma]_q$. Это отношение обычным образом распространяется на конечные мультимножества формул: для любых $v \in \mathcal{UF}$, $U, V \in FM(\mathcal{UF})$ если $v \to_R V$, то $U \cup \{v\} \to_R U \cup V$. В случае системы из единственного правила ρ вместо $\to_R$ пишем просто $\to_\rho$. Кроме отношения редукции введем также отношение $(\to_R)_b$, называемое отношением ветвления, порождаемым R, которое определяется следующим образом:

$$(\to_R)_b = \{(A, A') | \exists W(A \to_R W \wedge A' \in W)\}.$$

Это отношение потребуется для доказательства ряда свойств СПФ.

Пример 2.9. Пусть ρ — правило из примера 22, формула $A = p(h(z), z)$ экзистенциально связана. Возьмем в качестве НОУ $h(y)$ и $h(f(x))$ подстановку

$$\theta_1 = (x \to x', y \to f(x'))$$

и в качестве НОУ $h(y)$ и $h(g(x))$ подстановку

$$\theta_2 = (x \to x', y \to g(x')).$$

Правило ρ применимо к формуле A в позиции $q = 1$. В этом случае

$$\sigma = (y \to z).$$

Тогда

$$\phi_1 = (x \to x'', y \to f(x''), z \to f(x''), x' \to x'')$$

— тривиальный унификатор θ_1 и σ,

$$\phi_2 = (x \to x'', y \to g(x''), z \to g(x''), x' \to x'')$$

— тривиальный унификатор θ_2 и σ. Следовательно,

$$p(h(z), z) \to_\rho \{p(x'', f(x'')), p(x'', g(x''))\}.$$

Как и для обычных СПТ, для СПФ можно определить понятия нетеровости, нормальной формы, редекса и согласования с частичным порядком.

Определение 2.10. Пусть $U, V, W \in \mathcal{FM}(\mathcal{UF})$. V называется нормальной формой U относительно R, если $U \to_R^* V$ и для любого W неверно, что $U \to_R W$. Выражение t называется редексом R, если найдется правило $\rho = (\mathcal{B}, s) \in R$ такое, что $t = s\sigma$ для некоторой подстановки σ и для любой ветви $\pi = p|l \to r$ этого правила подстановки θ_π и σ тривиально унифицируемы, где θ_π — НОУ l и s. СПФ R называется нетеровой, если $\to_R$ нетерово. СПФ R согласована с частичным порядком $\succ$, если $\to_R \subseteq \succ$.

3. КОРРЕКТНОСТЬ СИСТЕМ ПЕРЕПИСЫВАНИЯ ФОРМУЛ

Свойство правила переписывания формул переписывать мультимножества формул из некоторого класса мультимножеств формул только в эквивалентные мультимножества формул называется корректностью правила относительно этого класса мультимножеств формул. Введем формальное

Определение 3.1. Пусть $\rho = (\mathcal{B}, s)$ — правило переписывания формул, $\pi = p|l \to r$ — ветвь ρ, $\bar{x}_\pi$ — множество переменных ветви ρ. Основа $\mathcal{B}$ называется корректной, если для любой ветви π формула

$$\forall \bar{x}_\pi (p \Rightarrow l = r)$$

истинна. Пусть $\mathcal{U} \subseteq \mathcal{FM}(\mathcal{UF})$. Правило ρ называется корректным относительно $\mathcal{U}$, если для любых $U, V \in \mathcal{U}$ из $U \to_\rho V$ следует, что U эквивалентно V.

Достаточные условия корректности для СПФ дает следующая

Теорема (корректности) 3.2. Пусть $\rho = (\mathcal{B}, s)$ — правило переписывания формул, $\pi = p|l \to r$ — ветвь ρ, θ_π — НОУ s и l,

$$\bar{x} = \cup_\pi (\mathcal{V}ar(l) \cup \mathcal{V}ar(s)),\ \bar{y}_\pi = \mathcal{VR}ange(\theta_\pi),$$
$$\bar{n}_\pi = (\mathcal{V}ar(p) \cup \mathcal{V}ar(r)) \setminus (\mathcal{V}ar(l) \cup \mathcal{V}ar(s)).$$

Если $\mathcal{B}$ корректна и формула

$$\forall \bar{x} \vee_\pi \exists \bar{y}_\pi \exists \bar{n}_\pi (\bar{x} = \bar{x}\theta_\pi \wedge p\theta_\pi)$$

истинна, то ρ корректно относительно множества $\mathcal{FM}(\mathcal{UF})$.

Доказательство. Пусть $A \in \mathcal{FM}(\mathcal{UF})$, $q \in \mathcal{P}(A)$, $\sigma \in \mathcal{S}$, $s\sigma = A_q$, σ и θ_π тривиально унифицируемы, ϕ_π — тривиальный унификатор σ и θ_π,

$$\bar{z} = \mathcal{V}ar(A),\ \bar{z}_1 = \mathcal{V}ar(A_q),\ \bar{z}_2 = \bar{z} \setminus \bar{z}_1,\ \bar{u}_\pi = \mathcal{VR}ange(\phi_\pi),$$

В случае $\exists$-формул достаточно доказать, что формулы

$$\exists \bar{z} A \quad \text{и} \quad \vee_\pi \exists \bar{z}_1 \exists \bar{u}_\pi \exists \bar{n}_\pi ((p\sigma \wedge A[r\sigma]_q)\phi_\pi)$$

эквиваленты.

Докажем сначала следующее

Предложение 3.3. Формула

$$\forall \bar{z}_2 \vee_\pi \exists \bar{u}_\pi \exists \bar{n}_\pi (\bar{z}_2 = \bar{z}_2\phi_\pi \wedge p\sigma\phi_\pi)$$

истинна, если выполнены условия теоремы.

Доказательство. Из условий теоремы имеем, что формула

$$\forall \bar{x} \vee_\pi \exists \bar{y}_\pi \exists \bar{n}_\pi (\bar{x}\sigma = \bar{x}\theta_\pi \wedge p\theta_\pi)$$

истинна. Из определения ϕ_π следует истинность формулы

$$\forall \bar{z}_2 \forall \bar{x} \forall \bar{y}_\pi \exists \bar{u}_\pi (\bar{x}\sigma = \bar{x}\theta_\pi \Rightarrow \bar{z}_2 = \bar{z}_2\phi_\pi \wedge \bar{y}_\pi = \bar{y}_\pi\phi_\pi \wedge \bar{x} = \bar{x}\phi_\pi)$$

для любой ветви π. Тогда формула

$$\forall \bar{x} \forall \bar{z}_2 \vee_\pi \exists \bar{y}_\pi \exists \bar{n}_\pi \exists \bar{u}_\pi (\bar{z}_2 = \bar{z}_2\phi_\pi \wedge \bar{y}_\pi = \bar{y}_\pi\phi_\pi \wedge \bar{x} = \bar{x}\phi_\pi \wedge p\theta_\pi)$$

также истинна. В контексте этой формулы выводима следующая цепочка эквивалентных формул:

$$p\theta_\pi \sim p\theta_\pi\phi \sim p\sigma\theta_\pi.$$

Тогда формула

$$\forall \bar{z}_2 \vee_\pi \exists \bar{u}_\pi \exists \bar{n}_\pi (\bar{z}_2 = \bar{z}_2\phi_\pi \wedge p\sigma\phi_\pi)$$

истинна. $\square$

Продолжим доказательство теоремы. Пусть формула $\exists \bar{z} A$ истинна. Из доказанного предложения и условий теоремы следует истинность формулы

$$\vee_\pi \exists \bar{z}_1 \exists \bar{u}_\pi \exists \bar{n}_\pi (A\phi_\pi \wedge p\sigma\phi_\pi).$$

В контексте этой формулы выводима следующая цепочка эквивалентных формул:

$$A_q\phi_\pi \sim s\sigma\phi_\pi \sim s\theta_\pi\phi_\pi \sim l\theta_\pi\phi_\pi \sim l\sigma\phi_\pi \sim r\sigma\phi_\pi.$$

Тогда формула

$$\vee_\pi \exists \bar{z}_1 \exists \bar{u}_\pi \exists \bar{n}_\pi ((p\sigma \wedge A[r\sigma]_q)\phi_\pi)$$

истинна.

Пусть теперь формула

$$\vee_\pi \exists \bar{z}_1 \exists \bar{u}_\pi \exists \bar{n}_\pi ((p\sigma \wedge A[r\sigma]_q)\phi_\pi)$$

истинна. В контексте этой формулы выводима следующая цепочка эквивалентных формул:

$$r\sigma\phi_\pi \sim l\sigma\phi_\pi \sim l\theta_\pi\phi_\pi \sim s\theta_\pi\phi_\pi \sim s\sigma\phi_\pi \sim A_q\phi_\pi.$$

Тогда формула

$$\vee_{\pi}\exists\bar{z}_1\exists\bar{u}_{\pi}\exists\bar{n}_{\pi}(A\phi_{\pi})$$

истинна, что влечет истинность формулы $\exists\bar{z}A$

Случай $\forall$-формул аналогичен $\square$

Пример 3.4. Вернемся к примеру 29. В этом случае

$$\bar{x} = (x, y),\ \bar{y}_1 = \bar{y}_2 = (x'),\ \bar{n}_1 = \bar{n}_2 = ().$$

Тогда согласно теореме 32 правило ρ корректно, если основа $\mathcal{B}$ корректна, т. е. формулы

$$\forall x h(f(x)) = x \text{ и } \forall x h(g(x)) = x$$

истинны и, кроме того, истинна формула

$$\forall x\forall y(\exists x'(x = x' \wedge y = f(x')) \vee \exists x'(x = x' \wedge y = g(x'))).$$

4. КОРРЕКТНОСТЬ СИСТЕМ ПЕРЕПИСЫВАНИЯ ФОРМУЛ В МНОГОСОРТНЫХ ТЕОРИЯХ

До сих пор рассматривались только односортные алгебраические системы или односортные теории как классы односортных алгебраических систем. В качестве семантики сортов возьмем теоретико-множественную семантику, в которой сорта рассматриваются как подмножества универсума в односортной логике. Такое задание семантики позволяет, с одной стороны, остаться в рамках односортной логики, а с другой — использовать более гибкую систему сортов, в частности подсорта, чем в обычной многосортной логике. Системы переписывания формул очевидным образом обобщаются на этот случай. В аксиоматических многосортных теориях удобно отделять аксиомы, определяющие свойства функциональных и предикатных символов от аксиом, определяющих свойства сортов. Определим многосортные теории как пары (T, τ), где τ — теория сортов, T — теория, задающая свойства функциональных и предикатных символов. Пусть $:s$ — предикат такой, что $u : s$ означает, что выражение u имеет сорт s.

4.1. Сорта, определяемые обобщенными конструкторами

Часто встречающимся на практике классом теорий сортов являются теории сортов, определяемых конструкторами. Дадим формальное

Определение 4.1. Пусть $s, s_1, ..., s_n$ — попарно различные сорта. Будем говорить, что сорт s определяется конструкторами $c_1, ..., c_k$ над сортами $s_1, ..., s_n$, если сорта аргументов конструкторов входят в множество сортов $\{s, s_1, ..., s_n\}$, сорт s является сортом результата конструкторов и выполнена следующая аксиома исчерпывания для конструкторов:

$$\forall x \vee_{1 \le i \le k} \exists \bar{y}_i (x = c_i(\bar{y}_i)).$$

Пример 4.2. Сорт $list$ списков определяется конструкторами добавления элемента к началу списка $cons$ и пустого списка nil. Аксиома исчерпывания для него принимает вид:

$$\forall x : list \exists y : elem \exists z : list (x = cons(y, z) \vee x = nil).$$

Сорта, определяемые конструкторами, допускают следующее очевидное обобщение, идея которого заключается в том, чтобы дать более общий вид аксиомы исчерпывания, а именно:

$$\forall \bar{x} \vee_{1 \le i \le l} \exists \bar{y}_i (p_i \wedge \bar{x} = \bar{t}_i),$$

где p_i — некоторые бескванторные формулы, $\bar{t}_i$ — векторы термов. Совокупность этих формул и термов образуют обобщенные конструкторы для обобщенных аксиом исчерпывания. Представим информацию о них более компактно. Пусть

$$\bar{p} = (p_1, ..., p_l),$$
$$\bar{\theta} = (\theta_1, ..., \theta_l),$$

где θ_i — подстановки вида $(\bar{x}_i \to \bar{t}_i)$. Тогда пара $(\bar{p}, \bar{\theta})$ задает всю информацию об обобщенном конструкторе. Дадим формальное определение обобщенных конструкторов и определяемых ими теорий сортов.

Определение 4.3. Пара $\bar{c} = (\bar{p}, \bar{\theta})$ называется обобщенным конструктором, если

$$\mathcal{D}om(\theta_i) \cap \mathcal{VR}ange(\theta_j) = \emptyset$$

и

$$\mathcal{D}om(\theta_i) \cap \mathcal{V}ar(p_j) = \emptyset$$

для любых $1 \le i, j \le l$. Теория τ называется теорией сортов, определяемой обобщенными конструкторами $\bar{c}_1, ..., \bar{c}_k$, если аксиомами этой теории являются в точности аксиомы исчерпывания для обобщенных конструкторов $\bar{c}_1, ..., \bar{c}_k$, имеющие следующий вид:

$$\forall \bar{x}^j \vee_{1 \le i \le l^j} \exists \bar{y}^j p_i^j \wedge \bar{x}^j = \bar{x}^j \theta_i^j,\ 1 \le j \le k,$$

где

$$\begin{array}{l} \bar{c}_j = (\bar{p}^j, \bar{\theta}^j), \\ l^j \text{ — длина вектора } \bar{p}^j, \\ \bar{x}^j = \cup_{1 \le i \le l^j} \mathcal{D}om(\theta_i^j), \\ \bar{y}^j = \cup_{1 \le i \le l^j} (\mathcal{VR}ange(\theta_i^j) \cup \mathcal{V}ar(p_i^j)). \end{array}$$

В дальнейшем будем использовать термин "конструктор"вместо термина "обобщенный конструктор". Рассмотрим примеры теории сортов, определяемых конструкторами.

Пример 4.4. Пусть теория сортов τ задает информацию о том, что сорт s является подсортом сорта s'. Тогда τ можно задать как теорию сортов, определяемую конструктором

$$\bar{c} = ([true], [(x \to y)]).$$

Действительно, аксиома исчерпывания для этого конструктора

$$\forall x : s \exists y : s'(x = y)$$

выражает свойство, что сорт s является подсортом сорта s'.

Пример 4.5. Пусть теория сортов τ задает информацию о том, что любые два элемента x и y сортов s и s' связаны формулой $p(x, y)$. Тогда τ можно задать как теорию сортов, определяемую конструктором

$$\bar{c} = ([p(u, v)], [(x \to u, y \to v)]).$$

Действительно, аксиома исчерпывания для этого конструктора

$$\forall x : s \forall y : s' \exists u : s \exists v : s'(p(u, v) \wedge x = u \wedge y = v)$$

выражает требуемое свойство. Таким образом можно, например, формализовать понятие зависимого типа, используемого в системе верификации PVS [47].

4.2. Корректность систем переписывания формул в теориях сортов

В многосортных теориях с выделенными подтеориями сортов, определяемых обобщенными конструкторами, проверку корректности СПФ можно упростить. При выполнении определенного синтаксического условия корректности СПФ в такой подтеории сортов достаточные условия корректности СПФ сводятся к условным равенствам, а не представляют $\forall\exists$-формулы, как в описаннной в предыдущем пункте теореме корректности.

Определение 4.6. Пусть теория сортов τ определяется конструкторами

$$\bar{c}_1, ..., \bar{c}_k,$$

R — СПФ. R называется корректной в τ, если для любого правила

$$\rho = (\{p_i | l_i \to r_i | 1 \le i \le n\}, s) \in R$$

$(\bar{p}, \bar{\theta})$ — обобщенный конструктор теории τ (с точностью до переименования переменных), где $\bar{\theta} = (\theta_1, ..., \theta_n)$, θ_i — НОУ l_i и s.

Теорема 4.7. Пусть (T, τ) — некоторая теория с подтеорией сортов τ, определяемой обобщенными конструкторами. Тогда R корректна в T, если R корректна в τ и для любого правила $\rho \in R$ и для любой ветви $p|l \to r$ этого правила формула $p \Rightarrow l = r$ истинна в T.

Доказательство. Данная теорема является непосредственным следствием теоремы корректности □

Пример 4.8. Пусть

$$\rho = (pred(succ(x)) \to x, pred(0) \to 0, pred(y))$$

— правило переписывания формул. Покажем, что оно корректно в теории (T, τ), где

$$T = \{pred(succ(x)) = x, pred(0) = 0\},$$

τ определяется конструктором

$$\bar{c} = ([true, true], [(y \to succ(x)), (y \to 0)]).$$

Правило ρ корректно относительно τ, так как НОУ образца $pred(y)$ и левых частей ветвей $pred(succ(x))$ и $pred(0)$ правила ρ имеют соответственно вид $(y \to succ(x))$ и $(y \to 0)$ и вместе образуют вторую компоненту конструктора $\bar{c}$, а первая его компонента образуется опущенными условиями ветвей $true$. Поэтому согласно теореме достаточно показать, что формулы $pred(succ(x)) = x$ и $pred(0) = 0$ истинны в теории T, что верно, так как на самом деле эти формулы — аксиомы теории T.

5. УНИФОРМНЫЕ СИСТЕМЫ

Рассмотрим специальный класс СПФ — униформные системы, для которых подстановки ϕ_π — переименования. Таким образом, системы этого класса отличаются от обычных СПТ только разбором случаев, а для систем с единственной ветвью сводятся к обычным СПТ с точностью до переименования переменных, поэтому ряд свойств СПТ переносится на униформные системы. Покажем, что проблема нетеровости неразрешима для униформных систем и для них имеется связь нетеровости с порядками упрощения, аналогичная связи в СПТ.

Определение 5.1. Пусть $*$ обозначает логическую связку $\Rightarrow$, $\vee$ или $\wedge$. Отображение Dec называется декомпозицией выражений, если

$$\begin{array}{l} Dec(true) = \emptyset,\ Dec(false) = \emptyset; \\ Dec(t = t') = \{t, t'\} \text{ для любых } t, t' \in \mathcal{T}; \\ Dec(\neg A) = Dec(A),\ Dec(A * B) = Dec(A) \cup Dec(B). \end{array}$$

Пусть $\succ$ — строгий частичный порядок. R убывает относительно $\succ$, если для любого правила $\rho \in R$ для любой ветви $\pi = p|l \to r$ этого правила $Dec(l) \succ_m Dec(r) \cup Dec(p)$. Пусть ρ — правило переписывания формул, π — ветвь ρ. Ветвь π называется униформной, если $l = s$. Правило называется униформным, если униформны все ветви этого правила. СПФ R называется униформной, если она состоит только из униформных правил.

Теорема (нетеровости) 5.2. Проблема нетеровости неразрешима для униформных систем.

Доказательство. Будем доказывать от противного. Пусть существует алгоритм μ, который проверяет, является униформная система нетеровой или нет. Покажем, что этот алгоритм можно использовать и для проверки свойства нетеровости для СПТ, состоящих из единственного правила. Пусть $\rho = \{l \to r\}$ — правило переписывания термов. Возьмем униформное правило переписывания формул $\rho' = (\{l \to r\}, l)$. Пусть $\mathcal{FP} = \mathcal{UF} \times \mathcal{UF}$. Имеется следующая цепочка легко проверяемых тождеств:

$$\rho \text{ нетерово } \Leftrightarrow (\to_\rho)_{|\mathcal{FP}} \text{ нетерово;}$$
$$(\to_\rho)_{|\mathcal{FP}} \text{ нетерово } \Leftrightarrow (\to_{\rho'})_b \text{ нетерово;}$$
$$(\to_{\rho'})_b \text{ нетерово } \Leftrightarrow \rho' \text{ нетерово.}$$

Из этих тождеств следует, что ρ' нетерово тогда и только тогда, когда ρ нетерово. Значит, алгоритм μ можно использовать и для проверки свойства нетеровости для СПТ, состоящих из единственного правила, что противоречит неразрешимости этого свойства для таких систем [19]. $\square$

Так как униформные системы являются частным случаем СПФ, то имеем

Следствие 5.3. Проблема нетеровости неразрешима для систем переписывания формул.

Рассмотрим связь нетеровости униформных систем с порядками упрощения. Соответствующая связь между СПТ и порядками упрощения рассматривалась в [52].

Теорема 5.4. Пусть R — униформная СПФ. Униформная СПФ R нетерова, если найдется стабильный относительно подстановок порядок упрощения $\succ$ такой, что R убывает отосительно $\succ$.

Доказательство. Пусть

$$\succ_n = \{(u, v) | Dec(u) \succ_m Dec(v)\}.$$

Докажем сначала следующее

Предложение 5.5. Для любых $u, u' \in \mathcal{FM}(\mathcal{UF})$ найдется переименование μ такое, что

$$u(\to_R)_b u' \Rightarrow u \succ_n u'\mu.$$

Доказательство. Пусть

$$A[s\sigma]_q(\to_R)_b(p\sigma \wedge A[r\sigma]_q)\phi.$$

Достаточно доказать, что существует переименование μ такое, что

$$A[s\sigma]_q \succ_n (p\sigma \wedge A[r\sigma]_q)\phi\mu.$$

Согласно определению ϕ и свойству униформности найдется переименование τ такое, что $\phi = \sigma\tau$. Тогда

$$A\phi = A\sigma\tau = A\tau$$

и

$$(p\sigma \wedge A[r\sigma]_q)\phi = (p\sigma \wedge A[r\sigma]_q)\tau.$$

Так как R убывает относительно $\succ$,

$$Dec(l) \succ_m Dec(p) \cup Dec(r).$$

Согласно определению порядка упрощения $\succ$ стабильно относительно постановок и обладает свойством подтерма. Поэтому

$$Dec(A[s\sigma]_q) \succ_m Dec(p\sigma) \cup Dec(A[r\sigma]_p)$$

и при $\mu = \tau^{-1}$ выполнено

$$A[s\sigma]_q \succ_n (p\sigma \wedge A[r\sigma]_q)\phi\mu.$$

□

Продолжим доказательство теоремы. Будем доказывать от противного. Пусть R не нетерова. Тогда найдется бесконечная цепь

$$u_0(\rightarrow_R)_b u_1(\rightarrow_R)_b \ldots \, .$$

Согласно предложению найдется бесконечная последовательность

$$\mu_1, \mu_2, \ldots$$

переименований такая, что

$$u_{i-1} \succ_n u_i\mu_i$$

для каждого $i \geq 1$. Пусть

$$u'_0 = u_0$$

и

$$u'_i = u_i\mu_i\mu_{i-1}\ldots\mu_1$$

для каждого $i \geq 1$. Так как $\succ$ стабильно относительно подстановок, $\succ_n$ также стабильно относительно подстановок и, следовательно,

$$u'_{i-1} \succ_n u'_i$$

для каждого $i \geq 1$. Согласно определению порядка упрощения $\succ$ нетерово. Тогда $\succ_n$ также нетерово, что противоречит существованию бесконечной цепи

$$u_0 \succ_n u_1 \succ_n \ldots \, .$$

Следовательно, R нетерова. □

6. СИСТЕМЫ ЭЛИМИНАЦИИ АНАЛИЗАТОРОВ

Разобъем множество всех функциональных и предикатных символов на два непересекающихся множества $\mathcal{D}$ анализаторов и $\mathcal{C}$ конструкторов. Логические связки и логические константы также отнесем к конструкторам. Пусть $G \subseteq \mathcal{F}$. Обозначим $\mathcal{E}(G)$ и $\mathcal{T}(G)$ соответственно множество формул и термов, содержащих функциональные и предикатные символы только из множества G. Введем понятия, позволяющие анализировать структуру выражений относительно конструкторов и анализаторов.

Определение 6.1. Выражение u называется конструктивным, если $u \in \mathcal{E}(\mathcal{C})$. Подстановка σ называется конструктивной, если для любой переменной z выполнено $z\sigma \in \mathcal{T}(\mathcal{C})$. Выражение u называется линейным, если для любой переменной z

$$|\mathcal{O}(z, \mathcal{MV}ar(u))| \leq 1.$$

Подстановка σ называется линейной на $X \subseteq \mathcal{V}$, если для любых $x, y \in X$ терм $x\sigma$ — линейный и

$$x \neq y \Rightarrow \mathcal{V}ar(x\sigma) \cap \mathcal{V}ar(y\sigma) = \emptyset.$$

Пусть $q_1, q_2 \in \mathcal{P}(u)$, $q_2 \neq q_1$, q_2 — префикс q_1. Выражение u называется вложенным, если найдутся q_1, q_2 такие, что

$$root(u_{q_1}), root(u_{q_2}) \in \mathcal{D}.$$

Выражение u называется простым, если u не является вложенным. Выражение u называется вызовом, если $root(u) \in \mathcal{D}$. Пусть $C_m(u)$ обозначает мультимножество всех простых вызовов, входящих в u. Отображение μ_c называется мерой конструкторов, если

$$\mu_c(u) = \{|t| | t \in C_m(u)\}.$$

Отображение Dec_v называется декомпозицией переменных, если

$$Dec_v(u) = \cup_{t \in C_m(u)} \mathcal{MV}ar(t).$$

Все рассматриваемые ниже СПФ учитывают разбиение на конструкторы и анализаторы.

Определение 6.2. СПФ R называется конструктивной с множеством конструкторов $\mathcal{C}$ и множеством анализаторов $\mathcal{D}$, если для любого правила $\rho = (\mathcal{B}, s) \in R$, для любой ветви $\pi = p|l \to r$ этого правила s и l — простые вызовы.

Родственное понятие для СПТ рассматривалось в работе [41]. Все рассматриваемые в этом и следующих пунктах классы СПФ состоят из конструктивных систем, позволяющих элиминировать анализаторы. Идея элиминации анализаторов основана на просачивании анализаторов через конструкторы вплоть до переменных с последующей их элиминацией при помощи специально подобранной замены переменных. Что касается свойства нетеровости для выбранных классов СПФ, то в общем случае не все системы этих классов нетеровы, однако для всех классов предлагается стратегия, относительно которой все системы из них завершаются.

Определим теперь первый класс систем — системы элиминации анализаторов (СЭА).

Определение 6.3. Пусть $\pi = p|l \to r$ — ветвь правила ρ. Ветвь π называется просачиваемой, если π униформна, p, r просты и

$$Dec_v(l) \supseteq Dec_v(p) \cup Dec_v(r),\ \mu_c(l) >_m \mu_c(r),\ \mu_c(l) >_m \mu_c(p).$$

Ветвь π называется элиминирующей, если любой НОУ выражений l и s линеен на $\mathcal{V}ar(s)$, p, r конструктивны. Конструктивная СПФ R называется системой элиминации анализаторов, если любой простой вызов — редекс R и правила R состоят только из просачиваемых и элиминирующих ветвей.

Пример 6.4. Пусть СПФ

$$R = \{\ (\{f(h(x)) \to x\}, f(y)),\ f(h(x)) \to x,\ f(g(x)) \to g(f(x))\ \}.$$

R — конструктивная система с анализатором f и конструкторами g и h. Докажем, что R — система элиминации анализаторов.

Ветвь $\pi_1 = f(h(x)) \to x$ первого правила является элиминирующей, так как $p_1 = true$, $r_1 = x$ конструктивны и НОУ

$$\theta_1 = (x \to z, y \to h(z))$$

выражений $f(h(x))$ и $f(y)$ линеен на $\{y\}$.

Ветвь $\pi_2 = f(h(x)) \to x$ второго правила является просачиваемой, так как она униформна, $p_2 = true$, $r_2 = x$ просты и выполнены условия

$$\{x\} \supseteq_m \emptyset \cup \emptyset,\ \{3\} >_m \emptyset,\ \{3\} >_m \emptyset.$$

Ветвь $\pi_3 = f(g(x)) \to g(f(x))$ третьего правила является просачиваемой, так как она униформна, $p_3 = true$, $r_3 = g(f(x))$ просты и выполнены условия

$$\{x\} \supseteq_m \emptyset \cup \{x\},\ \{3\} >_m \emptyset,\ \{3\} >_m \{2\}.$$

Таким образом, ветви правил СПФ R являются или просачивающими или элиминирующими. Вид правил, входящих в R, также гарантирует, что любой простой вызов — редекс системы R. Поэтому R — система элиминации анализаторов.

Как уже было сказано, завершимость СЭА достигается при выборе определенной стратегии применения правил СПФ. Определим понятие стратегии применения СПФ и, в частности, стратегию "любой самый внутренний", относительно которой завершаются СЭА.

Определение 6.5. Пусть $\mathcal{R}$ — множество всех СПФ данной сигнатуры Σ. Функция

$$s : \mathcal{R} \to \mathcal{FM}(\mathcal{F}) \times \mathcal{FM}(\mathcal{F})$$

называется стратегией применения СПФ, если $s(R) \subseteq \to_R$ для любой $R \in \mathcal{R}$. Будем говорить, что система R завершается относительно стратегии s, если $s(R)$ — нетерово отношение.

Согласно стратегии "любой самый внутренний"правила системы R применяются только к таким редексам этой системы, которые не содержит редексов R в качестве собственных подвыражений. Связь между СЭА и этой стратегией устанавливает следующая

Теорема 6.6. Системы элиминации анализаторов завершаются относительно стратегии "любой самый внутренний".

Доказательство. Пусть R — система элиминации анализаторов, s_i — стратегия "любой самый внутренний". Чтобы доказать завершимость R относительно s_i, достаточно построить нетеровый порядок $\succ$ такой, что $s_i(R) \subseteq \succ$ для любой R. Введем некоторые вспомогательные определения и обозначения. Пусть $u \in \mathcal{E}$. Обозначим $C_e(u)$ мультимножество всех вложенных вызовов u. Отображение μ_a называется мерой анализаторов, если

$$\mu_a(u) = |C_e(u)|.$$

Отображение μ_v называется мерой переменных, если

$$\mu_v(u) = \{\mathcal{O}(x, Dec_v(u)) | \mathcal{O}(x, Dec_v(u)) \neq 0\}.$$

Тогда требуемый частичный порядок $\succ = \succ'_m$, где вспомогательный порядок $\succ'$ определяется на $\mathcal{E}$ следующим образом: для любых выражений u и v $u \succ' v$ тогда и только тогда, когда

$$(\mu_a(u), \mu_v(u), \mu_c(u))$$

лексикографически больше, чем

$$(\mu_a(v), \mu_v(v), \mu_c(v)),$$

с порядками $>$, $>_m$ и $>_m$ на первом, втором и третьем элементах кортежа соответственно. Так как замыкание нетерова порядка на мультимножествах и лексикографический порядок на кортежах с фиксированным числом элементов, порожденные нетеровыми порядками, снова являются нетеровыми, $\succ'$ и $\succ$ нетеровы. Проверка $s_i(R) \subseteq \succ$ сводится к рутинному разбору случаев и потому опускается. Разбор же при упрощенной формулировке теоремы приведен в [2] $\Box$

7. СИСТЕМЫ ЭЛИМИНАЦИИ АНАЛИЗАТОРОВ СО СТАТУСОМ

Еще одним классом систем переписывания формул, нетеровых относительно стратегии "любой самый внутренний", являются системы элиминации анализаторов со статусом, обобщающие системы элиминации анализаторов. Идея обобщения состоит в том, что при анализе структуры выражений относительно конструкторов и анализаторов учитываются не все аргументы анализаторов, а только специальным образом выбранные. Такой выбор осуществляется с помощью специальной функции, сопоставляющей каждому анализатору множество позиций выбираемых аргументов. Введем формальное

Определение 7.1. Функция $\mathcal{A} : \mathcal{D} \to \mathcal{FS}(\mathcal{N})$ называется аргументной, если $\mathcal{A}(f) \subseteq \{1, ..., \mathcal{A}r(f)\}$ для каждого $f \in \mathcal{D}$.

Зафиксируем некоторый аргументный статус $\mathcal{A}$ и рассмотрим, как изменятся понятия, введенные для СЭА, если учитывать этот статус.

Определение 7.2. Пусть $q_1, q_2 \in \mathcal{P}(u)$, $q_2 \neq q_1$, q_2 — префикс q_1. Выражение u называется вложенным, если найдутся q_1, q_2 такие, что

$$root(u_{q_1}), root(u_{q_2}) \in \mathcal{D}.$$

Выражение u называется простым, если u не является вложенным. Обозначим $C_m(u)$ мультимножество всех простых вызовов, входящих в u,

$I(t)$ — мультимножество аргументов вызова t, находящихся в позициях $\mathcal{A}(root(t))$. Отображение μ_c называется мерой конструкторов, если

$$\mu_c(u) = \{ \sum_{t' \in I(t)} |t'| | t \in C_m(u) \}.$$

Отображение Dec_v называется декомпозицией переменных, если

$$Dec_v(u) = \cup_{t \in C_m(u)} \cup_{t' \in I(t)} \mathcal{MV}ar(t').$$

Определим теперь системы элиминации анализаторов со статусом.

Определение 7.3. Ветвь π называется просачиваемой, если π униформна, p, r просты и

$$Dec_v(l) \supseteq Dec_v(p) \cup Dec_v(r),\ \mu_c(l) >_m \mu_c(r),\ \mu_c(l) >_m \mu_c(p).$$

Ветвь π называется элиминирующей, если любой НОУ θ выражений l и s линеен на $\mathcal{V}ar(s)$, p, r конструктивны и для любой $x \in \mathcal{V}ar(s)$

$$x\theta \notin \mathcal{V} \Rightarrow x \in \cup_{s' \in I(s)} \mathcal{V}ar(s').$$

Конструктивная СПФ R называется системой элиминации анализаторов, если любой простой вызов (не обязательно относительно $\mathcal{A}$!) — редекс R и правила R состоят только из просачиваемых и элиминирующих ветвей.

Пример 7.4. Пусть СПФ R состоит из правил

$$\begin{array}{l} (\{f(h(x), z) \to x\}, f(y, z)), \\ f(h(x), z) \to x, \\ f(g(x), z) \to f(x, g(z)). \end{array}$$

R — конструктивная система с анализатором f и конструкторами g и h. Докажем, что R — система элиминации анализаторов с аргументным статусом $\mathcal{A}$, равным $\{1\}$ для анализатора f.

Ветвь $\pi_1 = f(h(x), z) \to x$ первого правила является элиминирующей относительно $\mathcal{A}$, так как $p_1 = true$, $r_1 = x$ конструктивны, НОУ

$$\theta_1 = (x \to x', y \to h(x'), z \to z')$$

выражений линеен на $\{x, z\}$,

$$x\theta_1 = x' \in \mathcal{V}$$

и

$$z\theta_1 = z' \in \mathcal{V}.$$

Ветвь $\pi_2 = f(h(x), z) \to x$ второго правила является просачиваемой относительно $\mathcal{A}$, так как она униформна, $p_2 = true$, $r_2 = x$ просты относительно $\mathcal{A}$ и выполнены условия

$$\{x\} \supseteq_m \emptyset \cup \emptyset,\ \{2\} >_m \emptyset,\ \{2\} >_m \emptyset.$$

Ветвь $\pi_3 = f(g(x), z) \to f(x, g(z))$ третьего правила является просачиваемой относительно $\mathcal{A}$, так как она униформна, $p_3 = true$, $r_3 = f(x, g(z))$ просты относительно $\mathcal{A}$ и выполнены условия

$$\{x\} \supseteq_m \emptyset \cup \{x\},\ \{2\} >_m \emptyset,\ \{2\} >_m \{1\}.$$

Таким образом, ветви правил СПФ R являются или просачивающими или элиминирующими относительно $\mathcal{A}$. Вид правил, входящих в R, также гарантирует, что любой простой вызов — редекс системы R. Поэтому R — система элиминации анализаторов с аргументным статусом $\mathcal{A}$.

Теорема 7.5. Системы элиминации анализаторов со статусом завершаются относительно стратегии "любой самый внутренний".

Доказательство. Доказательство аналогично доказательству теоремы 66 с тем же нетеровым порядком $\succ$. $\square$

8. СИСТЕМЫ ЭЛИМИНАЦИИ АНАЛИЗАТОРОВ С БАЗОЙ ПОДСТАНОВОК

Третьим классом конструктивных систем, завершающихся относительно стратегии "любой самый внутренний", являются системы элиминации анализаторов с базой подстановок. Идея таких систем заключается в следующем:

1. Выбирается некоторый класс выражений — база подстановок, обладающий определенными свойствами, и при анализе структуры любого выражения подвыражения, попадающие в этот класс, не учитываются. Примером такого анализа структуры выражений является применение функций $\mathcal{V}ar_E$ и $||_E$, не учитывающих подвыражения, входящие в E.

2. Требуется, чтобы наиболее общий унификатор образца и левой части неуниформной ветви любого правила конструктивных систем заменял переменные образца на выражения только из базы подстановок.

В этом случае глобальные подстановки, применяемые к формулам при их переписывании, заменят переменные формул на выражения из базы подстановок и не повлияют на характеристики структуры формул, не учитывающие входящие в базу подстановок подвыражения.

Определим свойства базы подстановок (замкнутость и полноту) и некоторые дополнительные операции анализа выражений, не учитывающие входящие в базу подстановок подвыражения.

Определение 8.1. Пусть $E \subseteq \mathcal{E}$, $S \subseteq \mathcal{S}$. E называется замкнутым относительно S, если для любого $u \in E$, для любой $\sigma \in S$ имеет место $u\sigma \in E$. E называется полным относительно S, если множества E, $\mathcal{E} \setminus (E \cup \mathcal{V})$ замкнуты относительно S. E замкнуто (полно) относительно подстановок, если E замкнуто (полно) относительно $\mathcal{S}$. Отображение μ_c^E называется мерой конструкторов относительно E, если

$$\mu_c^E(u) = \{|t|_E | t \in C_m(u)\}.$$

Отображение Dec_v^E называется декомпозицией переменных относительно E, если

$$Dec_v^E(u) = \{\mathcal{MV}ar_E(t) | t \in C_m(u)\}.$$

Пример 8.2. Пусть

$$\begin{array}{l} E = \{h(x)\sigma | \sigma \in \mathcal{S}\}, \\ E' = \{h(x)\sigma | \sigma \text{ — конструктивная подстановка}\}. \end{array}$$

Тогда E полно относительно подстановок, E' полно относительно конструктивных подстановок.

Теперь можно определить системы элиминации анализаторов с базой подстановок.

Определение 8.3. Пусть E — некоторое множество выражений, полное относительно конструктивных подстановок, $\rho = (\mathcal{B}, s)$ — правило переписывания формул, $\pi = p|l \to r$ — ветвь ρ. Ветвь π называется просачиваемой относительно E, если π униформна, p, r просты и

$$Dec_v^E(l) \supseteq_m Dec_v^E(p) \cup Dec_v^E(r),\ \mu_c^E(l) >_m \mu_c^E(r),\ \mu_c^E(l) >_m \mu_c^E(p).$$

Ветвь π называется элиминирующей относительно E, если для любого НОУ θ выражений l и s

$$\{x\theta | x \in \mathcal{V}ar(s)\} \subseteq E,$$

p, r конструктивны. Конструктивная СПФ R называется системой элиминации анализаторов с базой подстановок E, если любой простой вызов — редекс R и правила R состоят только из просачиваемых и элиминирующих относительно E ветвей.

Пример 8.4. Пусть СПФ

$$R = \{\ (\{f(h(x)) \to x\}, f(y)),\ f(h(x)) \to x,\ f(g(x)) \to g(f(x))\ \}.$$

R — конструктивная система с анализатором f и конструкторами g и h. Докажем, что R — система элиминации анализаторов с базой подстановок

$$E = \{h(x)\sigma | \sigma \text{ — конструктивная подстановка}\}.$$

Ветвь $\pi_1 = f(h(x)) \to x$ первого правила является элиминирующей относительно E, так как $p_1 = true$, $r_1 = x$ конструктивны и для НОУ

$$\theta_1 = (x \to z, y \to h(z))$$

выражений $f(h(x))$ и $f(y)$ выполнено свойство

$$\{h(z)\} \subseteq E.$$

Ветвь $\pi_2 = f(h(x)) \to x$ второго правила является просачиваемой относительно E, так как она униформна, $p_2 = true$, $r_2 = x$ просты и выполнены условия

$$\{\emptyset\} \supseteq_m \emptyset \cup \emptyset,\ \{1\} >_m \emptyset,\ \{1\} >_m \emptyset.$$

Ветвь $\pi_3 = f(g(x)) \to g(f(x))$ третьего правила является просачиваемой относительно E, так как она униформна, $p_3 = true$, $r_3 = g(f(x))$ просты и выполнены условия

$$\{x\} \supseteq_m \emptyset \cup \{x\},\ \{3\} >_m \emptyset,\ \{3\} >_m \{2\}.$$

Таким образом, ветви правил СПФ R являются или просачивающими или элиминирующими относительно E. Вид правил, входящих в R, также гарантирует, что любой простой вызов — редекс системы R. Поэтому R — система элиминации анализаторов с базой подстановок R.

Выполнение свойства завершимости систем элиминации анализаторов относительно E обеспечивает следующая

Теорема 8.5. Системы элиминации анализаторов относительно E завершаются относительно стратегии "любой самый внутренний".

Доказательство. Пусть R — система элиминации анализаторов с базой подстановок E. Чтобы доказать завершимость R относительно стратегии s_i, достаточно построить нетеровый порядок $\succ$ такой, что $s_i(R) \subseteq \succ$ для любой R. Требуемый частичный порядок $\succ = \succ'_m$, где вспомогательный порядок $\succ'$ определяется на $\mathcal{E}$ следующим образом: для любых выражений u и v $u \succ' v$ тогда и только тогда, когда

$$(\mu_a(u), \mu_c^E(u))$$

лексикографически больше, чем

$$(\mu_a(v), \mu_c^E(v)),$$

с порядками $>$ и $>_m$ на первом и втором элементах кортежа соответственно. Так как замыкание нетерова порядка на мультимножествах и лексикографический порядок на кортежах с фиксированным числом элементов, порожденный нетеровыми порядками, снова являются нетеровыми, $\succ'$ и $\succ$ нетеровы. Проверка $s_i(R) \subseteq \succ$ сводится к рутинному разбору случаев и потому опускается. □

ЗАКЛЮЧЕНИЕ

Кроме полного и систематического изложения теории СПФ в работе получены следующие новые результаты:

– Изложена с единых позиций семантика СПФ.

– Введен специальный класс теорий сортов — теории сортов, определяемые обобщенными конструкторами, и исследована связь этого класса с корректностью систем переписывания формул.

– Уточнены и расширены такие классы СПФ, как системы элиминации анализаторов и системы элиминации анализаторов со статусом.

– Предложен новый класс конструктивных систем, завершающийся элиминацией анализаторов относительно стратегии "любой самый внутренний", — системы элиминации анализаторов с базой подстановок.

– Доказана неразрешимость проблемы нетеровости для униформных систем.

Можно выделить две задачи, решение которых способствовало бы применению предложенной техники на практике:

– разработку новых методов терминации СПФ и создание достаточно полной теории терминации для таких систем. Решение этой задачи позволило бы разрабатывать тотальные разрешающие процедуры, основанные на СПФ;

– реализацию модуля переписывания формул, основанного на СПФ, и апробация его на различных классах задач. Практическое использование этого модуля позволило бы выявить границы и недостатки предложенной техники переписывания формул. Предварительная версия такого модуля уже разработана в рамках системы проблемно-ориентированной верификации СПЕКТР [11] и в настоящее время проходит проверку на условиях корректности для различных классов программ.

СПИСОК ЛИТЕРАТУРЫ

1. Ануреев И.С. Интегрированные правила переписывания термов и их применение в автоматической верификации программ // Проблемы спецификации и верификации параллельных систем. — Новосибирск, 1995. — С. 185—213.
2. Ануреев И.С. Системы переписывания формул. — Новосибирск, 1997. — 22 с. — (Препр./РАН.Сиб.Отд-ние. ИСИ; №40).
3. Ануреев И.С. Упрощающие процедуры для типов данных, основанные на системах переписывания формул. — Новосибирск, 1998. — 38 с. — (Препр./РАН.Сиб.Отд-ние. ИСИ; №50).
4. Воробьев С.Г. Переписывающие методы доказательства теорем в импликативных над базисными теориях // Всесоюз. конф. по прикладной логике: Тез. докл. — Новосибирск: Ин-т математики СО РАН СССР, 1985. — С. 39—42.
5. Воробьев С.Г. О выразительной силе систем переписывания термов // Применение методов математической логики (Секция "Представление знаний и синтез программ"): Тез. докл. IV Всесоюз. конф. — Таллин: Ин-т кибернетики АН ЭССР, 1986. — С. 48—50.
6. Воробьев С.Г. О применении условных систем подстановок термов в верификации программ // Программирование. — 1986. — №4. — С. 3—14.
7. Воробьев С.Г. О встраивании разрешимых фрагментов арифметики в системы переписывания термов // Методы трансляции и конструирования программ. — Новосибирск: ВЦ СО АН СССР, 1986.
8. Воробьев С.Г. Системы условных редукций и их применение в проблемно-ориентированной верификации программ: Дис. канд. физ.-мат.наук:05.13.11. — Новосибирск, 1987. — 150 с.
9. Кучеров Г.А. Системы подстановок термов. — Новосибирск, 1985. — 46 с. — (Препр./СО АН СССР. ВЦ; №601).
10. Массер Д. Спецификация абстрактных типов данных в системе AFFIRM // Требования и спецификации в разработке программ. — М.: Мир, 1984. — С. 199—222.
11. Непомнящий В. А. Сулимов А. А. Проблемно-ориентированные базы знаний и их применение в системе верификации программ СПЕКТР // Изв. РАН. Сер. Теория и системы управления. — 1997. — №2. — С. 169—175.

12. Anureev I.S. A method for simplification procedures design based on formula rewriting system // Joint Bulletin of IIS and Computer Center. — 1998. — №8. — (To appear).
13. Bachmair L., Ganzinger H. Rewrite-based equational theorem proving with selection and simplification // J. of Logic and Computation. — 1994. — №4(3). — P. 1—31.
14. Buchberger B. Basic features and development of the Critical-pair/Completion procedure // Lect. Notes Comput. Sci. — 1985. — Vol. 202. — P. 1—45.
15. Dershowitz N. Computing with rewrite systems // Information and Control. — 1985. — Vol. 65. — P. 122—157.
16. Dershowitz N., Plaisted D.A. Conditional rewriting // Software Eng.Notes. — 1985. — Vol. 10, №4. — P. 55—59.
17. Dershowitz N., Plaisted D.A. Logic programming cum applicative programming // Proc. of the 2nd IEEE Symposi. on Logic Programming. — Boston, 1985. — P. 54—66.
18. Dershowitz N., Plaisted D.A. Equational programming // Machine Intelligence. — 1987. — №11. — P. 21—56.
19. Dauchet M. Simulation of Turing machines by a left-linear rewrite rule // Lect. Notes Comput. Sci. — 1989. — Vol. 355. — P. 109—120.
20. Drosten K. Towards executable specification using conditional axioms // Lect. Notes Comput. Sci. — 1984. — Vol. 166. — P. 85—96.
21. Echahed R. On completeness of narrowing strategies // Theor. Comput. Sci. — 1990. — №72. — P. 133—146.
22. Echahed R. Uniform narrowing strategies // Lect. Notes. Comput. Sci. 1992. — №632. — P. 259—275.
23. Fay M. First-order unification in equational theories // Proc. of the 4th Internati. Workshop on Automated Deduction. — Austin, 1979. — P. 161—167.
24. Fribourg L. A narrowing procedure for theories with constructors // Lect. Notes Comput. Sci. — 1984. — Vol. 170. — P. 259—281.
25. Fribourg L. SLOG: A logic programming language interpreter based on clausal superposition and rewriting // Proc. of the 2nd IEEE Symposi. on Logic Programming. — Boston, 1985. — P. 172—184.
26. Giovannetti E., Levi G., Moiso C., Palamidessi C. Kernel-LEAF: A logic plus functional language // J. Comput. and Syst. Sci. — 1991. — №42. — P. 139—185.
27. Goguen J.A., Meseguer J. EQLOG: Equality,types and generic modules for logic programming // Logic Programming: Functions, Relations and Equations. — Englewood cliffs: Prentice Hall, 1986. — P. 295—363.
28. Hanus M. Compiling logic programs with equality // Lect. Notes Comput. Sci. — 1990. — №456. — P. 387—401.
29. Hsiang J. Two results in term rewriting theorem proving // Lect. Notes Comput. Sci. — 1985. — Vol. 202. — P. 301—324.
30. Hsiang J. Refutational theorem proving using term rewriting systems // Artif. Intell. — 1985. — Vol. 25, №2. — P. 255—300.
31. Hsiang J., Dershowitz N. Rewrite methods for clausal and non-clausal theorem proving // Lect. Notes Comput. Sci. — 1983. — Vol. 154. — P. 331—346.
32. Hsiang J., Kirchner H., Lescanne P., Rusinowitch M. Automated theorem proving in the presence of equalities // J. Automat. Reseach. — 1992. — Vol. 14. — P.71—100.
33. Hußmann H. Unification in conditional-equational theories // Lect. Notes Comput. Sci. — Vol. 204. — 1985. — P. 543—553.
34. Huet G., Hullot J.M. Proofs by induction in equational theories with constructors

// J. Comput. System Sci. — 1982. — Vol. 25, No 2. — P. 239—266.
35. Huet G., Oppen D.C. Equation and rewrite rules. A survey // Formal language theory: Perspectives and Open Problems. — New York: Acad.Press, 1980. — P. 349—406.
36. Hullot J.M. Canonical forms and unification // Lect. Notes Comput. Sci. — Vol. 87. — 1980. — P. 318—334.
37. Jouannaud J.P. Confluent and coherent equational term rewriting systems: applications to proofs in abstract data types // Lect. Notes Comput. Sci. — 1983. — Vol. 159. — P. 269—283.
38. Kaplan S. Conditional rewrite rules // Theor. Comput. Sci. — 1984. — Vol. 33, №2-3. — P. 175—193.
39. Kaplan S. Simplifying conditional term rewriting systems: unification, termination and confluence // J. Symb. Comput. — 1987. — №4(3). — P. 295—334.
40. Kapur D., Narendran P. An equational approach to theorem proving in first order predicate calculus // Software Eng. Notes. — 1985. — Vol. 10, №4. — P. 63—66.
41. Klop J.W. Term rewriting systems // Handbook Logic Comput. Sci. — 1993. — Vol. 2. — P. 1—116.
42. Lescanne P. Term rewriting systems and algebra // Lect. Notes Comput. Sci. — 1984. — Vol. 170. — P. 166—174.
43. Martelli A., Montanari U. An efficient unification algorithm // ACM Trans. on Prog.Lang. and Syst. — 1982. — Vol. 4, №2. — P. 258—282.
44. Moreno-Navarro J.J., Rodrìguez-Artalejo M. BABEL: A functional and logic programming language based on constructor discipline and narrowing // Lect. Notes Comput. Sci. — 1989. — Vol. 343. — P. 223—232.
45. Navarro M., Orejas F. On the equivalence of hierarchical and non-hierarchical rewriting in conditional term rewriting systems // Lect. Notes Comput. Sci. — 1984. — Vol. 174. — P. 74—85.
46. Nutt W., Rèty P., Smolka G. Basic narrowing revisited // J. Symb. Comput. — 1989. — №7. — P. 295—317.
47. Owre S., Rushby J., ShankarvN. PVS: A prototype verification system // Lect. Notes Art. Intell. — 1992. — Vol. 607. — P. 748—752.
48. Paul E. Equational methods in first order predicate calculus // J. Symb. Comput. — 1985. — №1. — P. 7—29.
49. Peterson G.E., Stickel M.E. Complete sets of reductions for some equational theories // J. ACM. — 1981. — Vol. 28, No 2. — P. 233—264.
50. Remy J.-L., Zhang H. REVEUR 4: A system for validating conditional algebraic specifications of abstract data types // Proc. 6th Europ. Conf. on Artificial Intell. (ECAI-84). — , 1984. — P.563—572.
51. Rèty P. Improving basic narrowing techniques // Lect. Notes Comput. Sci. — 1987. — Vol. 256. — P. 228—241.
52. Steinbach J. Simplification ordering: history of results // Fundamenta Informaticae. — 1995. — Vol. 24, №1. — P.47—87.
53. Yamamoto A. Completeness of extended unification based on basic narrowing // Proc. of the 7th Logic Programming Conference. — Jerusalem, 1988. — P. 1—10.
54. You Y.H. Enumerating outer narrowing derivations for constructor based term rewriting systems // J. Symb. Comput. — 1989. — №7. — P. 319—343.
55. Zhang H., Kapur D. Unnecessary inferences in associative-commutative completion procedures // Mathe. Systems Theory. — 1990. — №23. — P. 175—206.

56. Zhang H., Remy J.-L. Contextual rewriting // Lect. Notes Comput. Sci. — Vol. 202. — P. 46—62.

И. С. Ануреев

ТЕОРИЯ СИСТЕМ ПЕРЕПИСЫВАНИЯ ФОРМУЛ

Препринт
54

Рукопись поступила в редакцию 17.06.98
Рецензент Ф. А. Мурзин
Редактор Л. А. Карева

Подписано в печать 26.06.98
Формат бумаги 60×84 1/16 Объем 2,2 уч.-изд.л., 2,4 п.л.
Тираж 100 экз.

Отпечатано на ризографе "AL Group", 630090, пр. Акад. Лаврентьева, 6