УДК 519.172

А.Б. ГОДЛЕВСКИЙ

ПРЕДИКАТНЫЕ ПРЕОБРАЗОВАТЕЛИ В КОНТЕКСТЕ СИМВОЛЬНОГО МОДЕЛИРОВАНИЯ ТРАНЗИЦИОННЫХ СИСТЕМ

Ключевые слова: *атрибутные транзиционные системы, символьное моделирование, предикатный преобразователь, базовый протокол.*

ВВЕДЕНИЕ

Одним из основных направлений на стыке информатики и программной инженерии являются формальные методы как математический базис построения качественных программных продуктов. Формальные методы предполагают рассмотрение программ на уровне моделей, языков спецификаций и ориентированы на выявление в программах ошибок, их верификацию на ранних этапах разработки. Верификационный подход в программировании заложен в работах Р. Флойда [1] и Т. Хоара [2], в которых подчеркивалась роль спецификаций как для состояний программ (понятие assertions у Р. Флойда), так и для переходов между состояниями («тройки Хоара»). В рамках этого направления были введены понятия слабейшего предусловия и сильнейшего постусловия, которые на уровне языка логики и символьных исчислений позволяли характеризовать состояния программы как предшествующего данному, так и следующему за ним, исходя из характеристики текущего состояния и спецификации перехода. Позднее Л. Лэмпорт [3] использовал эти понятия для верификации параллельных программ и построения для них инвариантов, основанных на предусловиях и постусловиях.

Новым толчком в развитии формальных методов стал подход, получивший название проверки на моделях (model checking), когда в рамках модели разрабатываемой программы анализируются всевозможные пути ее исполнения и проверяется выполнимость анализируемых свойств. Этот подход, который впервые, по-видимому, был изложен в [4], предполагает использование различных форм абстракции для моделирования программ и происходящих в них процессов. Программы и комплексы взаимодействующих программ рассматриваются также в рамках модели транзиционных систем, где основными понятиями являются понятие состояния системы, правила переходов на пространстве состояний, описание свойств процессов, которые происходят в системе.

В данной работе исследуется модель взаимодействующих агентов и сред, введенная А.А. Летичевским [5] и развитая затем в рамках парадигмы инсерционного программирования [6], когда активация и функционирование агента (программы) во взаимодействии с другими агентами и их общей средой рассматривается как функция погружения агентов в среду. Эта идея взаимодействующих агентов нашла затем воплощение в реализации системы VRS (Verification Requirement System), разрабатываемой под руководством А.А. Летичевского и в сотрудничестве с компанией Motorola. Моделируемые приложения представляются атрибутными транзиционными системами, состояния этих систем — формулами логики, построенными над множеством атрибутов. Правила переходов в моделируемых системах представлены базовыми протоколами, которые подобно тройкам Хоара состоят из предусловия, обуславливающего применимость протокола, специфицируемого действия и постусловия, описывающего преобразование при данном переходе. При символьном моделировании атрибутных транзиционных систем ключевую роль играет механизм преобразования формул, представляющих классы состояний системы, по сути, механизм построения сильнейшего постусловия. В работе этот механизм, называемый предикатным преобразователем (predicate transformer), описыва-

ется для случая, когда допускаются атрибуты функционального типа, к которым, в частности, относятся массивы. Рассматривается случай суперпозиции атрибутов, когда аргументами одних атрибутов могут быть другие, скалярного или функционального типа. Основной результат состоит в доказательстве свойства для описываемого предикатного преобразователя строить сильнейшее постусловие.

1. ПРОСТЫЕ АТРИБУТНЫЕ ТРАНЗИЦИОННЫЕ СИСТЕМЫ

Пусть заданы конечная сигнатура $\Sigma = (\Omega, \Pi)$ функциональных и предикатных символов, конечное множество простых (скалярных) атрибутов A. Символы из Σ характеризуются арностью, типами аргументов и типом значения. Атрибуту $a \in A$ соответствует определенный тип τ и домен D_τ возможных значений. Понятие терма определяется двумя правилами. Во-первых, термами являются символы атрибутов и константы из доменов, которые соответствуют атрибутам из A. Тип такого терма определяется как тип атрибута или домена. Во-вторых, для всякого n-арного функционального символа f и термов $t_1, \ldots, t_n$, типы которых соответствуют типам аргументов f, термом будет $f(t_1, \ldots, t_n)$, и тип этого терма соответствует типу значений f. Выражение вида $\pi(t_1, \ldots, t_n)$ назовем предикатным термом, если π — n-арный предикатный символ, $t_1, \ldots, t_n$ — термы соответствующих типов. Формула — это выражение, построенное посредством пропозициональных связок из предикатных термов. Интерпретацией функциональных и предикатных символов относительно семейства доменов $\{D_\tau\}_{\tau \in T}$ назовем отображением I, которое каждому функциональному символу f сопоставляет (частичное) отображение типа $D_1 \times \ldots \times D_n \to D$, где домены $D_1, \ldots, D_n$, D соответствуют типу f, а каждому предикатному символу π — отображение типа $D_1 \times \ldots \times D_n \to \{0,1\}$.

Простая атрибутная транзиционная система (ATS) — это четверка $S = (S, BP, \mathrm{Init}, \mathrm{Final})$, где S — множество состояний системы, BP — конечное множество ее правил перехода (называемых ниже базовыми протоколами, или просто протоколами), Init и Final — соответственно множества начальных и финальных состояний. Состояние s — это отображение $A \to D_1 \cup \ldots \cup D_k$, которое каждому атрибуту a сопоставляет значение из домена, соответствующего типу этого атрибута. Если s — состояние системы, t и F — терм и формула над множеством атрибутов A, то выражениями $s(t)$ и $s(F)$ будут обозначаться значения этих терма и формулы в состоянии s. Базовый протокол — это конструкция вида $u(a_1, \ldots, a_n) \to < bp > v(a_1, \ldots, a_n)$, где $a_1, \ldots, a_n$ — некоторое множество атрибутов из A, bp — имя протокола, $u(a_1, \ldots, a_n)$ — предусловие и $v(a_1, \ldots, a_n)$ — постусловие протокола. Предусловие — это формула над множеством атрибутов $\{a_1, \ldots, a_n\}$, постусловие — групповой оператор присваивания, состоящий из нескольких простых присваиваний вида $a_i := t_i$. Термы в правых частях присваиваний могут зависеть от атрибутов $a_1, \ldots, a_n$, перечисленных в предусловии, но не от иных атрибутов из множества A. Семантика групповых присваиваний предполагает, что для заданного состояния s транзиционной системы вначале вычисляются значения всех правых частей присваиваний. Затем они объявляются новыми значениями соответствующих атрибутов из левых частей присваиваний, в результате система переходит в новое состояние s'. Протокол bp применим к состоянию s, если его предусловие выполнимо на s и результатом применения протокола будет состояние s', полученное из s под действием присваиваний постусловия.

Трассой в системе S будем называть последовательность $s_0 \xrightarrow{bp_0} s_1 \xrightarrow{bp_1} \ldots$ $\ldots s_k \xrightarrow{bp_k} s_{k+1} \ldots$ состояний, в которой каждое последующее состояние получается из предыдущего в результате применения к нему соответствующего базового протокола. Будем говорить, что множество Final достижимо из Init, если существует трасса, ведущая от одного из начальных состояний к одному из финальных. В терминах достижимости состояний формулируется одно из основных свойств

транзиционных систем — свойство безопасности: система S называется безопасной, если Final недостижимо из Init. Содержательно это свойство означает, что система не может достичь «плохих» состояний с точки зрения приложений. Проверка свойства безопасности является трудной задачей, поскольку множество S может быть бесконечным или очень большим. С этой целью применяется символьное моделирование, когда состояния системы разбиваются на классы и рассматриваются переходы между этими классами, индуцированные протоколами исходной системы. Затем проверка свойства безопасности может осуществляться двумя путями. Если имеется предположение о недостижимости множества Final, то ставится цель — доказать инвариантность множества состояний, дополняющих Final до универсума состояний. Если же имеется обратное предположение, то осуществляется проверка достижимости на модели (model checking), т.е. в моделирующей системе, где состояниями являются классы исходных состояний и рассматриваются обобщения правил перехода. В обоих этих подходах ключевыми пунктами соответственно являются построение модели транзиционной системы и построение внутри нее правил перехода, адекватных правилам в исходной системе.

Здесь в качестве состояний модели рассматриваются формулы сигнатуры Σ над множеством атрибутов A, а правила переходов строятся посредством предикатного преобразователя (predicate transformer), обозначаемого $pt(F, bp)$, аргументы которого — это формула F и базовый протокол bp. Иногда предикатный преобразователь будем обозначать также $pt(F\&u, v)$, уточняя, что оператор постусловия v трансформирует как исходную формулу F, так и предусловие u базового протокола. Преобразователь pt действует в три шага:

- проверяется, выполнима ли формула $F(a_1, \dots, a_k)\&u(a_1, \dots, a_k)$; если не выполнима, то это значит, что протокол bp не применим ко всем состояниям, где истинна F, и, следовательно, на модели этот переход также не применим к формуле;
- если формула $F(a_1, \dots, a_k)\&u(a_1, \dots, a_k)$ выполнима, то она преобразуется под действием постусловия $v(a_1, \dots, a_k)$;
- преобразованная на предыдущем шаге формула проверяется на выполнимость и упрощается, если это возможно.

В работе акцент делается на втором шаге, рассматривается алгоритм построения формулы $pt(F, bp)$ и доказываются ее свойства, демонстрирующие адекватность формульной модели. Относительно первого шага мы абстрагируемся от деталей конкретных транзиционных систем, от особенностей, обусловленных сигнатурой алгебраической системы, но предполагаем, что имеется алгоритм для проверки выполнимости формул из класса с данной сигнатурой. Иными словами, мы предполагаем, что имеется некий решатель (off-the-shelf solver), осуществляющий эту проверку. Заметим лишь, что применению решателя предшествует замена атрибутов $a_1, \dots, a_k$ связанными переменными $x_1, \dots, x_k$, затем проверяется выполнимость логической $\exists$-формулы $(\exists x_1, \dots, x_k)F(x_1, \dots, x_k)\&u(x_1, \dots, x_k)$.

Третий шаг, упрощение формулы, также находится за рамками данной работы и упоминается здесь лишь для того, чтобы подчеркнуть, что в практической системе эквивалентные преобразования по упрощению формулы важны, в частности если рассматривается целая цепочка последовательных преобразований. В частности, это могут быть действия по элиминации экзистенциальных кванторов, которые появляются при преобразовании формул.

Поскольку в процессе преобразований формул будут появляться кванторы, расширим класс рассматриваемых формул. Расширенный класс включает в себя $\exists$-формулы, которые получаются в результате замены некоторых атрибутов, входящих в формулу F, символами переменных из некоторого алфавита X. Предполагается, что связанные переменные типизированы и принимают значения внутри соответствующих доменов D_i. Формула $(\exists x_1, \dots, x_k)F(x_1, \dots, x_k, a_1, \dots, a_k)$ называется

выполнимой на состоянии s, если существует подстановка $\sigma: X \to D$, $D = D_1 \cup \ldots \cup D_n$, такая, что при замене связанных переменных из X их значениями $\sigma(x)$ получается формула, выполнимая на состоянии s. Заметим, что связанные переменные могут появляться внутри формул, описывающих классы состояний, но не внутри термов, что находятся в правых частях присваиваний.

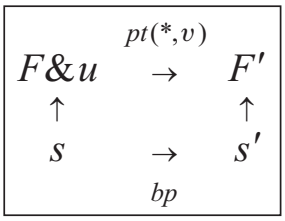

Рис. 1

Преобразование формул должно удовлетворять нижеследующей диаграмме (рис. 1), которая описывает идеальное моделирование исходной ATS символьной (формульной) моделью.

Здесь F — преобразуемая формула, $u \to < bp > \upsilon$ — базовый протокол, F' — результирующая формула, s' — состояние системы S, в которое она переходит под действием протокола bp из состояния s. Вертикальные стрелки на диаграмме отражают выполнимость формул $F \& u$ и F' в состояниях s и s'. Диаграмма отражает следующие свойства описываемого преобразователя:

а) если $s' = bp(s)$ и $F \& u$ выполнима в s, то $F' = pt(F \& u, \upsilon)$ должна быть выполнима в s';

б) если $F' = pt(F \& u, \upsilon)$ и F' выполнима в некотором s', то существует s такое, что $s' = bp(s)$ и $F \& u$ выполнима в s.

Предикатный преобразователь, обладающий такими свойствами, назовем *идеальным* и отметим, что это понятие преобразователя соответствует понятию сильнейшего постусловия (strongest precondition) из [2].

Построение формулы $pt(F \& u, \upsilon)$ осуществляется следующим образом:

• пусть $\{a_1, \ldots, a_n\}$ — множество всех атрибутов, находящиеся в левых частях присваивания υ, и пусть $a_1, \ldots, a_m$, $m \le n$, — все те из них, вхождения которых встречаются как в формуле $F \& u$, так и в термах правых частей присваиваний;

• пусть $x_1, \ldots, x_m$ — символы переменных, которые не встречались в формуле $F \& u$, тогда построим подстановку $\mathrm{Sub}^{x_1, \ldots, x_m}_{a_1, \ldots, a_m}$, которая, будучи примененной к формуле или терму, заменяет в ней все вхождения символа a_i на символ x_i;

• тогда $pt(F \& u, \upsilon) = (\exists x_1, \ldots, x_m)\ (G \& a_1 = t'_1 \& \ldots \& a_m = t'_m)$, где G — результат применения подстановки к формуле $F \& u$, когда часть атрибутов заменена переменными, и t'_i — результат применения этой же подстановки к терму t_i из правой части присваивания.

Теорема 1. Преобразователь $pt(*,*)$ является идеальным. Докажем свойства а) и б), определяющие свойство идеальности преобразователей. Формула $F \& u$ может содержать связанные переменные, но при рассуждениях их можно опустить, если полагать, что эти переменные заменяются одними и теми же значениями как в состоянии s, так и в s', а вновь вводимые связанные переменные $x_1, \ldots, x_m$ отличаются от уже существующих в формуле.

Пусть состояние s таково, что $s(F \& u)$. Это значит, что замена каждого атрибута a, входящего в формулу $F \& u$, на $s(a)$, элемент из D, обращает эту формулу в булевское значение 1. Зафиксируем эти значения и с учетом подстановки $\mathrm{Sub}^{x_1, \ldots, x_m}_{a_1, \ldots, a_m}$ выберем их для связанных переменных и для тех атрибутов, значения которых не изменились. Для тех же атрибутов, значения которых изменились оператором присваивания, в качестве новых значений выберем значения $s(t_i)$, которые совпадают с их значениями в состоянии $s' = bp(s)$. Теперь очевидно, что формула G будет выполнима в состоянии s' и тем самым свойство а) выполнено.

Пусть теперь s' — некоторое состояние, для которого формула G выполнима. Это значит, что существует отображение, сопоставляющее элементы доменов как атрибутам этой формулы, так и связанным переменным, появившимся в G при трансформации $F \& u$. Если элемент d_i сопоставляется связанной переменной x_i, то при построении

состояния s будем рассматривать его как значение атрибута a_i, встречающегося в левой части одного из присваиваний. Если же d_i — значение атрибута a_i, который не встречался в левых частях присваиваний из v, то зафиксируем это его значение для состояния s. По построению s и по тому, как строится формула G по $F \& u$, очевидно, что $F \& u$ будет выполнима на s и $s' = bp(s)$. Таким образом, доказано и второе свойство идеальности предикатного преобразователя $pt(*,*)$.

Теорема 1 не является новой и приводится здесь для того, чтобы ее идею использовать для более сложных предикатных преобразователей, действующих не только в простых атрибутных транзиционных системах.

2. ОБОБЩЕНИЯ ПРОСТЫХ АТРИБУТНЫХ ТРАНЗИЦИОННЫХ СИСТЕМ

Проблема построения более сложных предикатных преобразователей имеет несколько измерений, влияющих на возможность их построения:

i) использование атрибутов более сложного типа — функциональных атрибутов, частным случаем которых являются атрибутные массивы;

ii) спецификация постусловий — изменение атрибутов может задаваться не явно в виде присваивания, но неявно в виде констрейнта (формулы, ограничивающей возможные значения атрибута), определяющего целый спектр всех таких изменений. Возможна также комбинированная спецификация, когда изменение одной части атрибутов задается присваиваниями, а другой — посредством констрейнтов;

iii) использование конструкторов структур данных, позволяющих строить списки, очереди, деревья над значениями атрибутов;

iv) параметризация правил переходов (параметризованные правила позволяют вводить в формулу символьного состояния связанные переменные, которые будут учитывать определенные ограничения на значения атрибутов и строить более сложные формулы состояний).

Ниже детальнее рассмотрим предикатные преобразователи для ATS, в описании которых используются атрибуты функционального типа, но прежде сделаем несколько замечаний об обобщениях других типов. В случае (ii), когда формулы появляются в постусловии, можно выделить изменяемые неявным образом атрибуты и соответственно скорректировать преобразуемую формулу путем замены таких атрибутов связанными переменными, а далее рассматривать конъюнкцию преобразуемой формулы и формулы из постусловия. Случай (iii) интересен тем, что значения атрибутов могут копироваться, когда «старые» (измененные) значения атрибутов могут помещаться в списки или очереди и существовать одновременно с вновь присвоенными этим атрибутам значениями. При этом возможные коллизии, учитывающие повторное вычисление значений, должны отражаться в преобразованном формульном состоянии транзиционной системы. В случае (iv) используются базовые протоколы вида

$$(\forall x_1, \ldots, x_k)\, u(a_1, \ldots, a_m, x_1, \ldots, x_k) \rightarrow < bp > v(a_1, \ldots, a_m, x_1, \ldots, x_k),$$

где переменные $x_1, \ldots, x_k$ — параметры протокола. Их значения зависимы от значений атрибутов, входящих в формулу предусловия u, и затем от значений этих параметров будут зависеть новые значения атрибутов, определяемые постусловием v.

Отметим еще одно измерение, в котором рассматривается задача символьного моделирования: предикатные преобразователи могут использоваться не только при моделировании переходов от текущего состояния к последующему (forward simulation), но и при моделировании обратных переходов — от текущего состояния к предшествующему (backward simulation). И если рассматриваемые здесь преобразователи соответствуют вычислению сильнейшего постусловия, то преобразователи, моделирующие обратные переходы, соответствуют построению слабейшего предусловия.[1]

[1]Здесь понятия постусловия и предусловия отличаются от, фигурирующих в определении базовых протоколов, и соответствуют результатам прямого и обратного преобразований формул.

3. ATS С АТРИБУТАМИ ФУНКЦИОНАЛЬНОГО ТИПА

Атрибуты этого типа характеризуются арностью, типами аргументов (индексов) и типом значений. Если атрибут a имеет арность n, его аргументы принимают значения в доменах $D_1,\dots,D_n$, значения самого атрибута для наборов из $D_1\times\dots\times D_n$ относятся к домену D, то значениями атрибута являются функции из $D^{D_1\times\dots\times D_n}$. Объекты этого вида соответствуют также функциональным символам сигнатуры Ω и сопоставляются им интерпретацией I. Однако имеется различие между функциональными атрибутами и функциональными символами сигнатуры, которое состоит в том, что интерпретация функциональных символов остается неизменной при выполнении переходов в транзиционной системе, а значения функциональных атрибутов могут изменяться посредством применения базовых протоколов.

Использование функциональных атрибутов при спецификации базовых протоколов может вызвать противоречия в применении самих протоколов из-за возможных коллизий между левыми частями присваиваний в постусловии протокола. Если левые части двух операторов присваивания относятся к одному и тому же функциональному атрибуту и отличаются лишь разными аргументами, например $a(t_1,\dots,a_k)$ и $a(t'_1,\dots,a'_k)$, то следует быть уверенным, что для любого достижимого состояния транзиционной системы векторы значений аргументов будут различными. Атрибутные термы, у которых совпадает символ главного атрибута, будем называть подобными.

Предположим, что коллизии, связанные с подобием атрибутных термов в левых частях присваиваний, исключены. Это связано с тем, что при символьном моделировании процессов в ATS применение предикатных преобразователей как раз и нацелено на проверку достижимости всякого рода противоречивых состояний, и эта проверка определенным образом встраивается в процесс порождения формульных состояний модели. Проверка выполнимости для формул с функциональными атрибутами отличается тем, что при вычислении значения формул должен соблюдаться принцип функциональности, который выражается соотношениями вида $t_1=t_2\to a(t_1)=a(t_2)$, рассматриваемых Р. Шостаком [7]. С этой целью к формуле, проверяемой на выполнимость, добавляют такие соотношения для каждой пары входящих в нее подобных атрибутных термов. Как и ранее, для случая простых транзиционных атрибутных систем будем предполагать, что проблема проверки формул на выполнимость разрешима.

Пусть заданы формула $F(a_1,\dots,a_m)$ и базовый протокол $u(a_1,\dots,a_m)\to$ $\to<bp>v(a_1,\dots,a_m)$, зависящие от атрибутов из множества A. Пусть $a_1(t_1),\dots,a_k(t_k)$ — все левые части операторов присваивания в протоколе bp, и $t_1,\dots,t_k$ — векторы аргументов при соответствующих функциональных атрибутах. В формуле предикатного преобразователя $pt(F\&u,v)$, рассмотренной выше для случая скалярных атрибутов, атрибутные термы, подобные термам из списка $a_1(t_1),\dots,a_k(t_k)$, могли встречаться в формуле $F\&u$, а также в правых частях присваиваний. В случае с применением функциональных атрибутов атрибут, подобный данному, может встречаться также и в термах $t_1,\dots,t_k$, которые являются аргументами левых частей.

Пусть $a(t')$ — терм, подобный терму $a(t)$, одной из левых частей присваивания v, и такой, что он входит либо в $F\&u$, либо в правую часть одного из присваиваний, либо в один из аргументов одного из термов в левой части присваивания. Тогда значение этого терма $a(t')$ под действием присваивания $a(t):=tt$ может измениться или остаться неизменным в зависимости от того, имело ли место равенство $t'=t$ перед выполнением базового протокола. Чтобы разделить эти два случая, введем формулу $(t'=t\vee t'\neq t)$, которая, с одной стороны, тождественно равна единице, а с другой, позволит рассматривать отдельно трансформацию формул $F\&(t'=t)$ и $F\&(t'\neq t)$.

Построим конъюнкцию из всех формул вида $(t' = t \vee t' \neq t)$, образуемых для всех пар подобных атрибутных термов, один из которых относится к левой части присваивания, а другой является вхождением в один из вычисляемых термов. По построению эта конъюнкция равна единице. Раскроем скобки и построим д.н.ф., дизъюнкты которой содержат подформулы вида $(t'_{i1} = t'_{i1}) \& \ldots \& (t'_{im} = t'_{im}) \& \& (t'_{j1} \neq t'_{j1}) \& \ldots \& (t'_{jk} \neq t'_{jk})$, называемые здесь *классификаторами* и обозначаемые clf_i, где i — номер классификатора. Нас будут интересовать только выполнимые классификаторы, а невыполнимые, содержащие конъюнктивные взаимно противоречивые сомножители, будем исключать. Это возможно в силу предположения о том, что для формул с данной сигнатурой имеется разрешимая процедура проверки выполнимости. В классификаторе будем различать позитивную часть $(t'_{i1} = t'_{i1}) \& \ldots \& (t'_{im} = t'_{im})$, состоящую из равенств, и ограничительную $(t'_{j1} \neq t'_{j1}) \& \ldots \& (t'_{jk} \neq t'_{jk})$, состоящую из отрицаний равенств.

Поскольку классификаторы попарно не совместимы и в совокупности представляют тождественную формулу, преобразование формулы F будем сводить к преобразованию формул вида $F \& clf_i$. Опишем построение формулы $pt(F \& clf_i \& u, \upsilon)$, при этом акцент сделаем на функциональных атрибутах, поскольку эффект от присваиваний со скалярными атрибутами в левой части будет таким же как и в разделе выше.

Построение формулы $pt(F \& (t'_{i1} = t_{i1}) \& \ldots \& (t'_{im} = t_{im}) \& (t'_{j1} \neq t_{j1}) \& \ldots \ldots \& (t'_{jk} \neq t_{jk}) \& u, \upsilon)$, как и ранее, для формул со скалярными атрибутами проводится в два этапа:

• для термов образуется список свежих (fresh) переменных, связываемых затем кванторами существования, и строится подстановка, где каждому терму $a(t)$, являющемуся левой частью одного из присваиваний, сопоставляется переменная x;

• в преобразуемой формуле выделяются функциональные атрибуты, значения которых согласно равенствам из позитивной части классификатора будут изменены. Согласно семантике вычисление значений функций происходит в таком порядке: сначала вычисляются значения всех ее аргументов, а затем, исходя из этих значений, вычисляется значение самой функции. Поэтому обход преобразуемой формулы начинаем сверху вниз, отыскивая атрибутные термы, подобные термам в левых частях присваивания. Если находится атрибутный терм $a(t')$, то его список аргументов либо удовлетворяет $t' = t$, одному из равенств в позитивной части классификатора, либо отрицанию $t' \neq t$ одного из равенств ограничительной части последнего. В первом случае терм $a(t')$ заменяем связанной переменной, соответствующей терму $a(t)$, во втором обход преобразуемой формулы продолжается, начиная с аргументов терма $a(t')$.

Таким образом, получаем формулу для предикатного преобразователя:

$$pt(F \& u, \upsilon) = \bigvee_{clf_i} (\exists x_1, \ldots, x_k)\,(F' \& clf'_i \& u' \& (a_{i1}(\tau_{i1}) = tt'_{i1}) \& \ldots$$

$$\ldots \& (a_{ik}(\tau_{ik}) = tt'_{ik}), \qquad (1)$$

где F', clf'_i, u' — формулы, полученные из F, clf_i и u в результате замены некоторых атрибутных термов связанными переменными из множества $\{x_1, \ldots, x_k\}$. Применение этой же процедуры к $t_{i1}, \ldots, t_{ik}$, наборам аргументов в левых частях оператора υ, преобразует их к наборам $\tau_{i1}, \ldots, \tau_{ik}$, возможно, содержащим связанные переменные, и наконец, $tt'_{i1}, \ldots, tt'_{ik}$ — это трансформированные правые части операторов присваивания.

Теорема 2. Преобразователь $pt(*,*)$ идеальный для транзиционных систем с функциональными атрибутами.

Как и ранее, в случае теоремы 1, будем доказывать коммутативность диаграммы, представленной на рис. 1, т.е. докажем следующие два утверждения:

а) если $s' = bp(s)$ и $F\&u\&clf$ выполнима на s, то $F' = pt(F\&u,\upsilon)$ должна быть выполнима на s';

б) если $F' = pt(F\&u,\upsilon)$ и F' выполнима на некотором s', то существует s такое, что $s' = bp(s)$ и $F\&u$ выполнима на s.

Докажем а). Пусть $s(F\&u)$ и clf — тот единственный классификатор, который выполним на s, т.е. такой, что $s(clf)$. Для каждого присваивания $a(t_i) := t'_i$, входящего в υ, определим $c = s(t_i)$, набор аргументов функционального атрибутного терма $a(t_i)$, конкретизированных в состоянии s, и $d = s(t'_i)$ — конкретизированное значение правой части этого присваивания. Пусть x_i — символ переменной, используемый при построении формулы $pt(F\&u,\upsilon)$ для замены термов вида $a(t)$, набор аргументов которых входит в виде равенства $t = t_i$ в классификатор clf.

По построению $pt(F\&u,\upsilon)$ действие классификатора clf, истинного в состоянии s, таково, что всякий атрибутный подтерм $a(t)$, такой что равенство $t = t_i$ конъюнктивно входит в clf, будет заменен символом x_i, соответствующим атрибутному терму $a(t_i)$ из левой части одного из присваиваний в υ. Пусть

$$(\exists x_1,\dots,x_k)\,(F'\,\&\,clf'\,\&\,u'\,\&\,(a_1(\tau_1) = tt'_1)\&\dots\&\,(a_k(\tau_k) = tt'_k)) \qquad (2)$$

— дизъюнкт из (1), классификатор которого clf истинен в состоянии s. На множестве $X = \{x_1,\dots,x_k\}$ связанных переменных определим отображение $\sigma: X \to D$ такое, что $\sigma(x_i) = d_i = s(a(t_i))$. Заменим в (2) переменные из X значениями, которые им сопоставляет отображение σ. Тогда по построению $pt(F\&u,\upsilon)$ верно следующее:

• все атрибутные подтермы из $\sigma(F'), \sigma(u'), \sigma(clf')$ и $\sigma(\tau_{ij})$, подобные термам $a_1(\tau_1),\dots,a_k(\tau_k)$, в состоянии s будут иметь другие наборы конкретных аргументов, чем термы $a_1(\tau_1),\dots,a_k(\tau_k)$;

• значения $s(F)$ и $s(u)$ совпадают со значениями $s(\sigma(F'))$ и $s(\sigma(u'))$ соответственно;

• $s(clf)$ выполнимо согласно выбору классификатора из их множества;

• равенства вида $a_j(\tau_j) := tt'_j$ определяют новые значения функциональных атрибутов в состоянии s и их корректность следует из того, что рекурсия в них исключена в силу первого в этом перечислении утверждения.

В совокупности эти четыре утверждения позволяют заключить, что формула (2), а также $pt(F\&u,\upsilon)$, будет выполнима в состоянии s', которое отличается от s только значениями функциональных атрибутов $a_{i1},\dots,a_{ik}$ для наборов конкретных аргументов, соответственно равных $s(\sigma(\tau_1)),\dots,s(\sigma(\tau_k))$.

Докажем б). Пусть $s': A \to D$ и $\sigma: \{x_1,\dots,x_k\} \to D$ таковы, что формула $pt(F\&u,\upsilon)$ выполнима в состоянии $s'*\sigma$. В этой формуле выделим тот единственный классификатор clf, который выполняется при заданных s' и σ, а также соответствующий дизъюнкт, представляемый формулой (2). Каждая переменная x_i в (2) по построению заменяет значение атрибутного терма, чьи аргументы совпадают согласно выбранному классификатору с аргументами терма $a(t_i)$. Это замечание позволяет интерпретировать значения $d_i = \sigma(x_i)$ как значения атрибута a_i в состоянии $(s'*\sigma)t_i$.

Если в формуле $F'\,\&\,u'\,\&\,clf'$ или среди наборов термов $\tau_1,\dots,\tau_k$, аргументов функциональных атрибутов из левых частей присваивания υ, встречается атрибутный терм $a(t)$, то значение $(s'*\sigma)t$, конкретизация его аргументов в $s'*\sigma$, будет отличаться от значений $(s'*\sigma)t_i$ для $i = 1,\dots,k$. Действительно, по построению классификатор clf должен содержать либо равенство $t = t_i$, либо его отрицание. Процедура замены атрибутных подтермов для термов t и t_i на переменные из X сохраняет отношения = или ≠ между термами $a(t)$ и $a(t_i)$, поэтому вхождения атрибутного терма $a(t)$ при конкретизации $s'*\sigma$ не противоречит равенствам $a_i(t_i) = tt'_i$.

Последнее замечание позволяет заменить вхождения переменных из X их значениями, определяемыми отображением σ. Если далее определить значения функ-

циональных атрибутов a_i, $i = 1, \ldots, k$, для наборов аргументов t_i как $(s'^*\sigma)t_i$, то получим новое состояние *s*. Это состояние отличается от s' только значениями атрибутов a_i для конкретных наборов аргументов $(s'^*\sigma)t_i$ и, следовательно, формула $F\&u\&clf$ будет выполнимой в состоянии *s*. Нетрудно также понять, что состояния s и s' удовлетворяют соотношению $s' = bp(s)$, что и доказывает теорему 2.

Теорема 2 утверждает, что преобразователь pt является сильнейшим постусловием в смысле [2] для атрибутных транзиционных систем с функциональными атрибутами.

Заметим, что состояние s' может быть не единственным, где выполняется $pt(F\&u, \upsilon)$, и более того, оно зависит от выбора отображения σ, действующего на множестве связанных переменных X. Если таких состояний s', удовлетворяющих формуле $pt(F\&u, \upsilon)$, несколько, то соответственно будет столько же состояний *s*, на которых выполнима формула F.

ЗАКЛЮЧЕНИЕ

С точки зрения комбинаторной сложности формула $pt(F\&u, \upsilon)$ может оказаться громоздкой, поскольку количество классификаторов растет экспоненциально по отношению к числу пар подобных атрибутных термов. Однако в практическом плане таких пар подобных атрибутных термов, во-первых, может быть относительно немного и, во-вторых, большинство классификаторов может оказаться тождественно невыполнимыми формулами. Например, классификатор может содержать равенства $(1,2) = (t_1, t_2)$ и $(3,4) = (t_3, t_2)$, которые несовместимы, так как не удовлетворяются ни при каком значении терма t_2.

Даже при высокой комбинаторной сложности применение подобного предикатного преобразователя для символьного моделирования атрибутных транзиционных систем оправдано, если рассматривать приложения, где процессы в системе относительно короткие, но в каждой точке имеется большое число вариантов для продолжения процесса. В этом случае процесс может моделироваться на уровне формул $F_1, \ldots, F_n$, которые на каждом шаге будут сопоставляться с некоей целевой формулой *G*, что будет выражаться в проверке выполнимости конъюнкций этих формул с *G*. Если на каком-то шаге такая конъюнкция выполнима, т.е. существует состояние s^* такое, что $s^*(F_n \& G)$, то встает задача обратного моделирования, когда конструируется трасса, ведущая от некоего допустимого начального состояния к найденному критическому состоянию s^*.

СПИСОК ЛИТЕРАТУРЫ

1. Floyd R.W. Assigning meaning to programs, in proceedings of symposium on applied mathematics / J.T. Schwartz (Ed.), A.M.S. — 1967. — **19**. — P. 19–32.
2. Hoare C.A.R. An axiomatic basis for computer programming // Com. of the ACM. — 1969. — **12**, N 10. — P. 576–580,
3. Lamport L. *win* and *sin*: predicate transformer for concurrency // ACM Transl. on Program. Language and System (TOPLAS). — 1990. — **12**, N 3. — P. 396–428.
4. Clarke E.M., Emerson E.A. Design and synthesis of synchronization skeletons using branching time temporal logic // Logics of Programs: Workshop (Yoktown Heights, NY). — May 1981. — **131**. — P. 52–71.
5. Letichevsky A., Gilbert D. A model for interaction of agents and environments // Lecture Notes In Comput. Scie. — 1999. — **1827**. — P. 311–328.
6. Спецификация систем с помощью базовых протоколов / А.А. Летичевский, Ю.В. Капитонова, В.А. Волков и др. // Кибернетика и системный анализ. — 2005. — № 4. — С. 3–21.
7. Shostak R. A practical decision procedure for arithmetic with function symbols // J. of the Assoc. for Comput. Machinery. — 1979. — **26**, N 2. — P. 351–360.

Поступила 09.04.2010