Сибирский математический журнал
Ноябрь—декабрь, 2006. Том 47, № 6

УДК 004.89

ТЕОРИЯ СТРУКТУРНЫХ ФУНКЦИОНАЛЬНЫХ МОДЕЛЕЙ

В. Б. Новосельцев

Аннотация: Предлагается теория, подобная логическим формализмам и предназначенная для описания прикладных предметных областей. Теория основана на хорошо известной «теории вычислительных моделей» Тыугу и Минца, но дополнительно включает такой важный инструмент описания предметной области, как рекурсия. Предлагаемая теория является вполне адекватной для непроцедурных описаний широкого класса прикладных областей и в то же время обладает полиномиальными оценками и является полной. Данный формализм используется как теоретический базис для разработки систем, основанных на знаниях.

Ключевые слова: формальная теория, рекурсия, доказательство теорем, логический вывод, непроцедурное описание.

Введение

В работе предпринимается попытка построения формализма, призванного обеспечить возможность создания строгих и в то же время адекватных моделей для широкого круга предметных областей (ПО). Рассматриваемые исследования традиционно относят к области искусственного интеллекта. Прикладной стороной исследования является создание интеллектуальных программных комплексов широкого класса, опирающихся на дедуктивный подход в области менеджмента знаний. Термин «знание» здесь имеет ограниченную трактовку: знания представляются специальными строго определенными информационными структурами, допускающими автоматизированную обработку с использованием аппарата формальной логики. Представляемые результаты могут быть использованы при реализации различных информационных комплексов — от экспертных и когнитивных систем до инструментальных и CASE-оболочек. Для достижения этой цели мы отталкиваемся от хорошо разработанной теории вычислительных моделей [1–3].

Работа состоит из трех параграфов. В § 1 определяются используемые понятия и необходимые соглашения. В § 2 вводится понятие интерпретации С-модели и формулируется исчисление функциональных предложений для теории структурных моделей [4], а также исследуются некоторые свойства построенного исчисления. В § 3 описаны стратегия и базовый алгоритм установления выводимости предложений языка. Для алгоритма доказывается свойство корректности и устанавливаются сложностные характеристики.

§ 1. Базовые определения

Зафиксируем сигнатуру Σ.

Определение 1. Четверку $\Sigma = (\mathscr{A}, \mathscr{F}, \mathscr{P}, \mathscr{D})$, где $\mathscr{A}, \mathscr{F}, \mathscr{P}$ и $\mathscr{D}$ — не более чем счетные множества (элементарных) имен объектов, функциональных символов, селекторных символов и имен схем соответственно, назовем *сигнатурой.*

Считается, что выделено непустое конечное подмножество $\mathscr{E} \subset \mathscr{D}$ имен *примитивных* или *первичных* схем. Элементы множества $\mathscr{A}$ называются *именами объектов* или *объектами*, если это не приводит к двусмысленности. Связь объекта a со схемой S отражается записью $S(a)$. Если объект a связан со схемой S и $S \in \mathscr{E}$, то стандартная запись $S(a)$ заменяется записью a^S либо просто a, когда ссылка на схему не важна или очевидна из контекста. Истинностное значение синонимов a^S и $S(a)$ определяется следующим образом: $a^S = \textbf{\textit{И}}$ тогда и только тогда, когда $a|_{\mathbf{I}} \in S|_{\mathbf{I}}$, где $a|_{\mathbf{I}}$ и $S|_{\mathbf{I}}$ обозначают реализации (на интерпретации $\mathbf{I}$) величины и схемы соответственно.

Определение 2. Выражение вида

$$f : a_1 \dots a_n \to a_0, \tag{1}$$

где a_i, $i = \overline{0, n}$, — имена объектов, называется *функциональной связью* (ФС).

В записи (1) f — это имя ФС, a_i — аргументы ($i = \overline{1, n}$), a_0 — результат ФС. Неформально ФС трактуется как возможность вычисления значения величины a_0 по набору значений величин $a_1, \dots, a_n$ применением реализации отображения f.

Имена объектов при моделировании конкретной ПО формируются из элементов множества $\mathscr{A}$ и удовлетворяют следующему рекурсивному определению.

Определение 3. Пусть $a \in \mathscr{A}$, σ — имя величины, тогда $a, \sigma.a$ и $a.\sigma$ также являются именами величин.

В записи вида $\sigma.a$ фрагмент σ называется *префиксом. Длина* произвольного имени σ определяется числом вхождений элементов множества $\mathscr{A}$ (с учетом возможных повторений), из которых сформировано имя.

Одним из базовых является понятие схемы (непервичной, т. е. схемы конструируемого объекта).

Определение 4. Схема S объекта r определяется выражением вида

$$\begin{aligned} S(r) = r.(&S_{01}(a_{01}), \dots, S_{0N_0}(a_{0N_0}), \\ &if\, p_1 \supset S_{11}(a_{11}), \dots, S_{1N_1}(a_{1N_1}) \Box \dots \Box \\ &\quad p_k \supset S_{k1}(a_{k1}), \dots, S_{kN_k}(a_{kN_k}) fi \mid filter), \end{aligned} \tag{2}$$

где $S \in \mathscr{D} \setminus \mathscr{E}$, $r \in \mathscr{A}$. Для всех возможных значений индексов i, j символы $S_{ij} \in \mathscr{D}$ обозначают *собственные подсхемы* схемы S, $a_{ij} \in \mathscr{A}$ — ее *собственные величины*, $p_i \in \mathscr{P}$ — (возможно, параметризованные) выбирающие *селекторы.* В правой части (2) r называется *префиксом* схемы, участок до вертикальной черты — *заголовком*, фрагмент $if \dots fi$ — *вариантной* его *частью*, а остальная часть заголовка — *постоянной частью.* Символ «$\Box$» заимствован из нотации «охраняемых команд» [5] и в некотором смысле соответствует традиционному «*else_if* ». Иероглиф *filter* скрывает список *собственных* ФС схемы S. Считается, что префикс r может «проноситься» в скобки, так что $r.(S(a), \dots) \rightleftharpoons (r.S(a), \dots) \rightleftharpoons (S(r.a), \dots)$. Аналогичным образом префиксируются имена селекторов и отображений, а также имена величин, вовлеченных в формирование ФС из *filter*[1)].

Селекторы на интерпретации **I** получают истинностные (шкальные) значения и образуют полную систему, т. е. $\forall i, j,\ i \neq j,\ p_i|_{\mathbf{I}} \& p_j|_{\mathbf{I}} = \mathbf{Л}$ и $p_l|_{\mathbf{I}} \vee \dots \vee p_k|_{\mathbf{I}} = \mathbf{И}$, что гарантирует свойство детерминированности схемы. Синтаксическая конструкция (2) на интерпретации **I** определяет размеченное отношение в смысле [6][2)].

Можно отметить некоторую аналогию между введенным здесь понятием схемы и отношением в теории реляционных баз данных [7]. Используя это, мы иногда будем употреблять термины реляционного подхода такие, как «атрибут», «подсхема», «кортеж» и т. д. Укажем теперь требования, которым удовлетворяет набор *filter* схемы. Для этого прежде всего определим, какие имена величин порождаются заголовком схемы.

Определение 5. Пусть $S(r) = (\dots S_{ij}(r.a_{ij})\dots)$ — схема. Если $S_{ij} \in \mathscr{E}$, то $r.a_{ij}$ — имя величины схемы S. Если $S_{ij} \in \mathscr{D} \setminus \mathscr{E}$ и α — имя величины схемы S_{ij}, то $r.a_{ij}.\alpha$ — имя величины схемы S. В дальнейшем имена величин схемы S будем называть *именами* ее *атрибутов* или *атрибутами.*

Определение 6. Рассмотрим схему S. ФС $f : a_1, \dots, a_n \to a_0$ из *filter* называется *допустимой для схемы* S, если и только если $a_0, \dots, a_n$ — атрибуты схемы S. Схема называется *синтаксически правильной*, если *filter* содержит только допустимые ФС.

Ниже будут использоваться только синтаксически правильные схемы. Помимо этого будем считать, что ФС, содержащиеся в наборе *filter* схемы S, отвечают следующим требованиям:

- ФС содержит по крайней мере один собственный атрибут схемы;
- в наборе *filter* нет ФС, в которые вовлечены атрибуты из разных вариантных ветвей схемы S или атрибуты из вариантных частей ее подсхемы;
- длина любого атрибута, связанного ФС, не превышает двух.

Иногда величины предметной области, для которой строится модель, имеют смысл только при выполнении некоторого критерия. Так, о гипотенузе и катетах можно говорить лишь в контексте прямоугольного треугольника. Для отражения подобных фактов служит вариантная часть заголовка схемы — она указывает, когда определены те или иные атрибуты. В ряде случаев информацию об осмысленности атрибутов схемы важно иметь в наборе *filter*. Для этого вводится понятие атрибута с условием, или *у*-атрибута.

Определение 7. Факт наличия условия p у атрибута a отмечается записью a/p, а соответствующее выражение называется *у-атрибутом.*

Считается, что с атрибутами, определенными в общей части заголовка, связывается условие **И**. Если в записи *у*-атрибута a/p известно, что $p \doteq \mathbf{И}$, то $/p$ можно опустить.

В качестве примера схемы приведем описание понятия *член_ряда*, определяющего n-й член натурального ряда либо ряда Фибоначчи в зависимости от значения скаляра s. Символом «N» будем обозначать тождественное отобра-

[1)]Фиксация множества $\mathscr{E} \subset \mathscr{D}$ для конкретной ПО не является окончательной. При необходимости можно сделать доопределение «вниз», представив ранее первичный элемент в форме (2) и детализировать рассматриваемую ПО соответствующей модификацией множества $\mathscr{E}$. Требование конечности множества $\mathscr{E}$ при этом не нарушается.

[2)]Определение 4, на первый взгляд, предоставляется не более чем механизмом введения сокращения при описании ПО, однако использование этой дисциплины важно для введения естественного порядка на атрибутах модели.

жение:

$$\begin{aligned}
\text{\textit{член_ряда}}(r) = r.(&scalar(s), integer(x), integer(n),\\
&if\ (= s\,\text{«\textit{натуральный}»}) \supset integer(xn) \Box\\
&(= s\,\text{«\textit{фибоначчи}»}) \supset \text{\textit{число_фиб}}(xf)\ fi \mid\\
&N: n \to xn; N: xn \to x; N: n \to xf.n; N: xf.f \to x).
\end{aligned}$$

В этом примере неэлементарной является схема *число_фиб*, которая определяет член ряда Фибоначчи. Описание этой схемы будет приведено позднее, когда мы коснемся рекурсивных конструкций. Собственными атрибутами схемы *член_ряда* являются s, x, n, xn и xf. В схеме *число_фиб*, связанной с атрибутом xf, определены атрибуты n и f. Таким образом, в схеме *член_ряда* появляются префиксированные атрибуты $xf.n$ и $xf.f$. Для натурального ряда значение n-го члена совпадает с его номером n — этот факт отражается двумя первыми ФС схемы. Число Фибоначчи вычисляется по более сложным законам, определяемым схемой *число_фиб*. «Вход» в подсхему *число_фиб* схемы *член_ряда* и «выход» из нее осуществляется посредством двух последних ФС набора *filter* схемы *член_ряда*.

Определим понятие структурной функциональной модели.

Определение 8. *Структурной* (С-) *моделью* называется конечная совокупность (описаний) схем $M = (S_1, \ldots, S_m)$, где каждая S_i, $i = \overline{1, m}$, является элементарной или задана в соответствии с определением 4[3)].

Определение 9. С-модель M является (*синтаксически*) *замкнутой*, если и только если для каждого ее элемента $S_i \in M$ схемы, встречающиеся в определении S_i, элементарны либо определены в описании С-модели M.

Описывая С-модель для предметной области, мы строим специальную (прикладную) теорию, в которой можно решать, в частности, проблему установления выводимости предложений соответствующего языка. Здесь, главным образом, рассматриваются проблема синтеза реализующих алгоритмов (программ) и сложностные характеристики синтеза этих программ.

Постановка задачи в С-модели носит непроцедурный характер — в ней указываются лишь исходные и искомые атрибуты некоторой схемы. Пользователем модели может выступать не только человек, но и любой функциональный агент, авторизованный в модели. Предполагается, что пользователи модели функционально согласованы с требованиями, заданными вышеприведенными определениями.

Содержательно постановка задачи на модели есть задание на построение схемы алгоритма, реализующего требуемые вычисления. Схема алгоритма извлекается из (конструктивного) доказательства соответствующей теоремы существования [4]. Схема становится алгоритмом (программой) в результате задания интерпретации модели.

Определение 10. *Задачей в* С-*модели* M называют тройку $T = (A_0, X_0, S)$, где A_0 и X_0 — наборы имен соответственно исходных и искомых величин, а S — схема С-модели M, в которой определены эти имена.

Для дальнейшего нам понадобится понятие развертки схемы.

Определение 11. Пусть $S(r) = r.(\ldots S_{ij}(a_{ij}), \cdots \mid filter)$ — схема, и $S_{ij} \notin \mathscr{E}$. *Разверткой схемы S на атрибуте a_{ij}* будем называть выражение, получающееся в результате (а) подстановки в заголовок исходной схемы на место

[3)] Можно заметить сходство определяемого здесь понятия модели и библиотеки классов в популярном «объектно-ориентированном» подходе.

$S_{ij}(a_{ij})$ заголовка схемы S_{ij} и (б) присоединения к набору *filter* схемы S ФС схемы S_{ij}. Все добавленные в схему S в результате такого действия атрибуты, имена селекторов и отображений модифицируются префиксом $r.a_{ij}$.

Процесс построения развертки на атрибуте может повторяться многократно, пока имеются атрибуты, связанные с неэлементарными схемами.

Определение 12. Под *полной разверткой* S С-*модели* понимается объект, полученный из схемы S, у которой в результате последовательности разверток на атрибутах в заголовке остаются только атрибуты, связанные с элементарными схемами.

Полная развертка, вообще говоря, не является схемой в смысле определения 4, однако понятие схемы нетрудно переопределить с тем, чтобы оно оставалось корректным для развертки на атрибуте и полной развертки. Для этого необходимо в выражении (2) разрешить появление нескольких $if \dots fi$-участков и потребовать, чтобы единственным видом их пересечения было строгое вложение (как оно трактуется в структурных языках программирования).

При построении С-модели допускаются возможность рекурсивных определений схем. В общем случае рекурсивная конструкция должна содержать по меньшей мере одну цепочку определений вида

$$S_1 = (\dots[S_2]\dots), S_2 = (\dots[S_3]\dots), \dots, S_k = (\dots[S_1]\dots),$$

обеспечивающую участие в дефиниции некоторого понятия ссылки на себя. Схемы S_i, $i = \overline{1,k}$, будем называть *рекурсивными.* Ясно, что полная развертка рекурсивной схемы не определена — процесс ее построения не завершается. Тем не менее существуют стратегии, обеспечивающие эффективное установление выводимости формул в предложенном формализме. Для теории функциональных моделей имеют место важные теоремы. Первая из них фиксирует положение класса С-моделей (без рекурсии) в общем классе моделей Диковского [8], а вторая гарантирует свойство эффективной распознаваемости предложенной теории.

Теорема 1. *Полная развертка любой схемы S нерекурсивной* С-*модели M определяет некоторую модель Диковского, если M составлена только из строго детерминированных схем.*

Доказательство. Рассмотрим схему.

$$\begin{aligned} S(r) = r.(&S_{01}(a_{01}), \dots, S_{0N_0}(a_{0N_0}),\\ &if\, p_1 \supset S_{11}(a_{11}), \dots, S_{1N_1}(a_{1N_1}) \square \dots \square\\ &\quad p_k \supset S_{k1}(a_{k1}), \dots, S_{kN_k}(a_{kN_k}) fi \mid filter). \end{aligned}$$

Напомним, что атрибутам схемы S естественным образом приписываются условия **И**, p или $-p$ в зависимости от местоположения в заголовке. В развертке исходной схемы S может появиться фрагмент (префикс «$r.$» опускается)

$$\dots if\ p \supset \dots if\ a_{st}.q \supset \dots a_{st}.S_{qr}(b_{qr}), \dots, fi \dots fi.$$

В этом случае атрибуту $a_{st}.b_{qr}$ приписывается условие $p \& a_{st}.q$. Поскольку полная развертка схемы в НДС-модели конечна, возможно произвести переименование и заменить сложные имена (уникальными) простыми. При этом требование счетности соответствующих элементов сигнатуры не нарушается. Для завершения доказательства достаточно сделать последний шаг. Набор $filter$ развертки содержит теперь различные ФС вида $f : c_1/p_1, \dots, c_n/p_n \to c_0/p_0$.

Припишем каждой ФС условие $P_f \rightleftharpoons p_0 \& p_1 \& \ldots \& p_n$. Соберем все ФС, правые части которых (результаты) совпадают, и сформируем для каждого результата детерминированное уравнение $\mu(c_0)$ вида $c_0 = if\ P_{f_1} \to f_1 \Box P_{f_2} \to f_2 \Box \ldots \Box P_{f_n} \to f_n\ fi$. Очевидно по построению, что (i) $n > 0$; (ii) $P_{f_1}, \ldots, P_{f_n}$ попарно различны; (iii) для каждого атрибута c имеется не более одного $\mu(c)$; (iv) детерминированных уравнений конечное число. $\Box$

А. Я. Диковским показано, что при выборе семантики, в которой не используется внешняя управляющая память для задания условий в детерминированных уравнениях, проблема синтеза в Д-вычислительных моделях является $\boldsymbol{NP}$-полной. В С-моделях алгоритмы, решающие проблему синтеза, обладают полиномиальными характеристиками.

Теорема 2. *Принадлежность* $M = (S_1, \ldots, S_m)$ *классу рекурсивных детерминированных* С-*моделей устанавливается конструктивно за конечное время, пропорциональное* $O(|M^4|) = C \times m^4$.

Примером рекурсивной схемы может служить определение n-го элемента ряда Фибоначчи:

$$
\begin{aligned}
&\textit{число\_}\text{фиб}(r) = r.(\textit{integer}(n), \textit{integer}(f),\\
&\qquad\qquad\qquad if(= n\ «0») \supset \textit{integer}(ff)\Box\\
&\qquad\qquad\qquad\quad (= n\ «1») \supset \textit{integer}(sf)\Box\\
&\qquad\qquad\qquad\quad (> n\ «1») \supset \textit{число\_фиб}(nf), \textit{число\_фиб}(nnf) fi\ |\\
&\quad N : «1» \to ff; N : sf \to f; N : «1» \to ff; N : ff \to f;\\
&\quad \textit{subtract} : n, «1» \to nf.n; \textit{subtract} : n, «2» \to nnf.n; \textit{sum} : nf.f, nnf.f \to f).
\end{aligned}
$$

§ 2. Интерпретация С-модели

Интерпретация (семантика) С-модели вводится стандартным образом.

Определение 13. *Интерпретация* $\mathbf{I}$ С-модели M задает: (а) для каждой элементарной схемы $S \in \mathscr{E}$ непустое множество (первичный тип) $S\ |_{\mathbf{I}}$; (б) для каждого функционального символа $f \in \mathscr{F}$ — отображение $f\ |_{\mathbf{I}}$: $S_1\ |_{\mathbf{I}} \times \cdots \times S_n|_{\mathbf{I}} \to S_0|_{\mathbf{I}}$; (в) для каждого селектора $p \in \mathscr{P}$ — булевское отображение $p|_{\mathbf{I}}$: $S_1\ |_{\mathbf{I}} \times \cdots \times S_n|_{\mathbf{I}} \to \{\mathbf{И}, \mathbf{Л}\}$.

В результате задания интерпретации каждой неэлементарной схеме сопоставляется отношение, т. е. множество наборов, удовлетворяющих некоторым условиям. Наборы принадлежат декартову произведению множеств, сопоставленных подсхемам схемы $\boldsymbol{S}$. Элементы наборов поименованы атрибутами заголовка. Если в заголовке схемы присутствует альтернативная часть, то наборы отношения выбираются из размеченного декартова произведения [6]. Разметку определяют селекторы альтернативной части. Термин «тип атрибута» служит сокращением для выражения «множество, сопоставленное схеме, с которым связан атрибут». В отличие от *первичного* типа такой тип мы будем называть (*непервичным* или просто) *типом*, наборы типа — *кортежами*, а их именованные элементы — *компонентами*.

Отображения, которые сопоставляются ФС схемы в результате задания семантики С-модели, определяют соотношения между компонентами кортежей. Выполнение этих соотношений указывает *принадлежность* кортежа типу. С компонентами кортежей связываются реализации условий, которые приписаны соответствующим этим компонентам *у*-атрибутам.

Для удобства и большей компактности рассуждений вводится ряд дополнительных соглашений, не имеющих принципиального характера.

Используется только двухальтернативная форма вариантной части из (2): $if\ p \supset \dots ; \dots fi$.

Вводится понятие *предложения вычислимости* (ПВ) вида $F : A \to X$, являющееся сокращающим обозначением для $\{F_{x_i} : A \to x_i\}$, где F_{x_i} — определяемые ниже программные термы, формирующие F, а $X \rightleftharpoons \cup\{x_i\}$.

Определение 14. Рассмотрим С-модель $M = (\dots S \dots)$. Пусть

$$a_1/p_1, \dots, a_n/p_n, \quad x_1/q_1, \dots, x_m/q_m$$

— y-атрибуты развертки S,

$$\Psi \rightleftharpoons F : a_1/p_1, \dots, a_n/p_n \to x_1/q_1, \dots, x_m/q_m$$

— предложение вычислимости, а r — кортеж, определяемый заголовком развертки при некоторой интерпретации $\mathbf{I}$. Будем говорить, что $\Psi|_{\mathbf{I}}$ *имеет смысл* для кортежа r, если $p_i|_{\mathbf{I}} = \boldsymbol{И}$, $i = \overline{1,n}$, и $q_j|_{\mathbf{I}} = \boldsymbol{И}$, $j = \overline{1,m}$.

Определение 15. Будем говорить, что ПВ $\Psi \rightleftharpoons F : A \to X$ *удовлетворяет схеме* S, если при любой интерпретации $\mathbf{I}$ тип $S|_{\mathbf{I}}$ не содержит двух кортежей, для которых $\Psi|_{\mathbf{I}}$ имеет смысл, таких, что их компоненты совпадают для всех атрибутов множества A, но не совпадают по крайней мере для одного атрибута множества X. При этом для произвольного кортежа r, который принадлежит отношению $S|_{\mathbf{I}}$, результатом применения $F|_{\mathbf{I}}$ к компонентам r, именованным атрибутами множества A, является набор компонент, именованных атрибутами множества X. Все ФС схемы удовлетворяют ей по определению.

В исчислении имеется три сорта термов.

Определение 16. (1) Термы *первого сорта* (a-термы) представляют собой упорядоченные списки атрибутов, имена которых строятся из элементов $\mathscr{A}$. (2) Термы *второго сорта* (y-термы) строятся из термов первого сорта и селекторных символов, принадлежащих множеству $\mathscr{P}$, согласно следующей рекурсивной процедуре. Пусть $p^{(n)} \in \mathscr{P}$ — селектор, реализуемый n-местной булевской функцией, A — n-элементный терм первого сорта такой, что его i-й атрибут связан с той же схемой, что и i-й аргумент p; P и Q — термы второго сорта; α — некоторый атрибут; «$-$» и «&» — символы отрицания и конъюнкции соответственно. Тогда перечисленные выражения являются термами второго сорта (других термов второго сорта нет): $\alpha.p(A), -P, P\&Q$. (3) Наконец, программные (*пр*-) термы *третьего сорта* строятся из термов всех сортов по следующим правилам. Пусть P — терм второго сорта; f, g, h — функциональные символы (элементы множества $\mathscr{F}$); F_1, F_2, F_3 — термы третьего сорта и имеются вхождения символа g в F_3; α — некоторый атрибут; $F_3[h/g]$ — обозначение замены вхождения g на h в терме F_3. Тогда $\alpha.h : \dots \to \dots$ — терм третьего сорта; F_1, F_2 — терм, построенный с применением оператора композиции;

$$if\ P\ then\ F_1\ else\ F_2\ fi$$

— терм (оператор ветвления);

$$h = if\ P\ then\ F_1\ else\ F_3[h/g]\ fi$$

— терм (оператор рекурсии). Других термов третьего сорта нет.

В рамках введенных соглашений можно использовать некоторый *базовый* набор правил вывода (заметим, что термы, реализующие функциональные связи, имеют традиционную для императивных языков программирования форму и соответственно трактовку):

(0) схема аксиом $\vdash N : A \to A$,

(1) $\dfrac{\vdash F : A \to X, Z}{\vdash F : A \to X}$ (правило сужения),

(2) $\dfrac{\vdash F : A \to X, Z/P \quad \vdash f : Z \to x/p}{\vdash F; f : A, Z/P \to X, Z, x/P\&p}$ (правило суперпозиции),

здесь P — конъюнкция (шкальных) условий достижимости атрибутов из Z, а p — условие допустимости x (условие достижимости — это условие, при котором в процессе построения доказательства обеспечивается достижимость некоторого атрибута, а условие допустимости — это условие, при котором атрибут имеет смысл в описании),

(3) $\dfrac{\vdash F_1 : A \to X, x/Q\&P \quad \vdash F_2 : A \to X, x/Q\& -p}{\vdash if\ p\ then\ F_1\ else\ F_2\ fi : A \to X, x/Q}$ (правило ветвления),

(4) $\vdash F_1 : A \to X, x/Q\&p$ (правило рекурсии).

$$\frac{g_1 : n_{a_1}^{k_1}(A) \to n_{a_1}^{k_1}(X), \ldots, g_s : n_{a_s}^{k_s}(A) \to n_{a_s}^{k_s}(X) \vdash F_2 : A \to X, x/Q\& -p}{h = if\,p\ then\ F_1 else\ F_2[h/g_1]\ldots[h/g_s]\ fi : A \to X/Q}$$

Для произвольной модели ПО приведенные правила обеспечивают весьма эффективный вывод сложностью не более, чем 3-я степень от длины описания исходной модели. Можно также утверждать, что описанная теория обладает свойствами эффективной разрешимости и полноты.

Построенное исчисление будем называть *исчислением* SR (structural recursive calculi).

Для исчисления SR справедливы следующие утверждения.

Теорема 3. *Исчисление* SR *является корректным относительно понятия* ПВ, *удовлетворяющего схеме* S.

Доказательство. Утверждение теоремы означает следующее: если ПВ $\Phi \rightleftharpoons F : A \to X$ выведено с помощью правил SR на основании информации об исходной схеме S модели M, то независимо от интерпретации $\mathbf{I}$ для любых двух кортежей типа $S|_{\mathbf{I}}$ таких, что $\Phi|_{\mathbf{I}}$ имеет для них смысл, компоненты, соответствующие атрибутам A и X в этих кортежах, совпадают и связаны соотношением, задаваемым $\Phi|_{\mathbf{I}}$.

Рассмотрим поочередно схему и все правила и покажем, что они позволяют выводить из ПВ, удовлетворяющих исходной схеме, только те ПВ, которые также ей удовлетворяют.

(0) Очевидно. Независимо от интерпретации $\mathbf{I}$ в $S|_{\mathbf{I}}$ не может быть двух кортежей таких, что в них присутствуют одни и те же компоненты $A|_{\mathbf{I}}$, и кортежи одновременно совпадают и не совпадают по этим компонентам.

(1) ПВ $\Phi \rightleftharpoons F : A \to X, Y$ — *антецедент правила* — удовлетворяет схеме S. Это означает, что любые два кортежа, для которых $\Phi|_{\mathbf{I}}$ имеет смысл, совпадающие по $A|_{\mathbf{I}}$, совпадают также по всему набору $X|_{\mathbf{I}}$, $Y|_{\mathbf{I}}$. Следовательно, эти кортежи совпадают и по набору $X|_{\mathbf{I}}$. Поскольку весь набор $X|_{\mathbf{I}}$, $Y|_{\mathbf{I}}$ является результатом применения отображения $F|_{\mathbf{I}}$ к $A|_{\mathbf{I}}$, то $X|_{\mathbf{I}}$ получается применением того же $F|_{\mathbf{I}}$ к $A|_{\mathbf{I}}$ с последующей проекцией на атрибуты X.

(2) (*Правило суперпозиции*). Пусть опять посылками правила являются ПВ, удовлетворяющие схеме S. Рассмотрим два кортежа s и t, для которых реализации посылок имеют смысл, причем s и t совпадают по компонентам $A|_{\mathbf{I}}$. Предположим, что правило (2) некорректно, это означает, что s и t не совпадают по какой-либо компоненте из набора $X|_{\mathbf{I}}$, $Z|_{\mathbf{I}}$, $x|_{\mathbf{I}}$. Несовпадение по фрагменту из $X|_{\mathbf{I}}$, $Z|_{\mathbf{I}}$ противоречит тому, что $F : A \to X, Z$ удовлетворяет схеме S, т. е. s и t совпадают, в частности, по $Z|_{\mathbf{I}}$, которые вычисляются применением $F|_{\mathbf{I}}$ к $A|_{\mathbf{I}}$ (с выполнением условия достижимости $P|_{\mathbf{I}}$). Последнее означает, что s и t должны совпадать и по $x|_{\mathbf{I}}$, поскольку $f : Z \to x$ удовлетворяет схеме S. Отметим, что в качестве второй посылки правила суперпозиции всегда берется ФС из

(развертки) исходной схемы, именно поэтому p — это условие допустимости x. Условие, при котором $f|_{\mathbf{I}}$ обеспечивает возможность вычисления $x|_{\mathbf{I}}$ по $Z|_{\mathbf{I}}$, не может быть более слабым, чем условие, при котором вычислены $Z|_{\mathbf{I}}$, и условие, при котором в $S|_{\mathbf{I}}$ допускается присутствие $x|_{\mathbf{I}}$ — оно определяется конъюнкцией $P\&p$. Набор $X|_{\mathbf{I}}$, $Z|_{\mathbf{I}}$, $x|_{\mathbf{I}}$ вычисляется последовательным применением $F|_{\mathbf{I}}$ к $A|_{\mathbf{I}}$ и затем $f|_{\mathbf{I}}$ к соответствующим результатам предыдущего отображения, т. е. суперпозицией отображений. Этим показано, что правило (2) исчисления SR корректно.

(3) (*Правило ветвления*). Предположим, что ПВ из заключения правила не удовлетворяет исходной схеме, если посылками являются ПВ, удовлетворяющие этой схеме. Введем обозначения для ПВ из формулировки правила:

$$\Phi_1 \rightleftharpoons F_1 : A \to X, x/Q\&p, \quad \Phi_2 \rightleftharpoons F_2 : A \to X, x/Q\& - p,$$

$$\Phi \rightleftharpoons if\ p\ then\ F_1\ else\ F_2\ fi : A \to X, x/Q.$$

Пусть задана интерпретация $\mathbf{I}$. Рассмотрим два кортежа s и t таких, что для них (обоих одновременно) имеет смысл либо $\Phi_1|_{\mathbf{I}}$, либо $\Phi_2|_{\mathbf{I}}$. Очевидно, что в каждом из этих случаев для s и t имеет смысл и $\Phi|_{\mathbf{I}}$. Пусть s и t совпадают по компонентам $A|_{\mathbf{I}}$, и пусть для них имеет смысл $\Phi_1|_{\mathbf{I}}$. Если наше предположение о том, что ПВ из заключения правила не удовлетворяет схеме, является верным, то s и t должны не совпадать по какой-либо компоненте из $X|_{\mathbf{I}}$, $x|_{\mathbf{I}}$, а это противоречит первой посылке правила. В рассматриваемом случае программный терм из заключения правила вырождается до F_1, кортежи s и t, совпадая по $X|_{\mathbf{I}}$, $x|_{\mathbf{I}}$, совпадают и по компонентам $X|_{\mathbf{I}}$, $x|_{\mathbf{I}}$, а соотношение между $A|_{\mathbf{I}}$ и $X|_{\mathbf{I}}$, $x|_{\mathbf{I}}$ определяется реализацией терма $if\ p\ then\ F_1\ else\ F_2\ fi$ при выполнении p. Второй случай (для s и t имеет смысл $\Phi_1|_{\mathbf{I}}$) рассматривается аналогично.

Корректность правила ветвления исчисления SR показана.

(4) Наконец, исследуем вопрос корректности правила введения рекурсии. Введем обозначения:

$$\Phi_1 \rightleftharpoons F_1 : A \to X, x/Q\&p, \quad \Phi_2 \rightleftharpoons F_2 : A \to X, x/Q\& - p,$$

$$\phi_1 \rightleftharpoons g_1 : n^{k_1}_{\alpha_1}(A) \to n^{k_1}_{\alpha_1}(X); \dots \phi_s \rightleftharpoons g_s : n^{k_s}_{\alpha_s}(A) \to n^{k_s}_{\alpha_s}(X),$$
$$\Phi \rightleftharpoons h = if\ p\ then\ F_1 else\ F_2[h/g_1]\dots[h/g_s]fi : A \to X/Q.$$

Отметим, что, хотя в правиле (4) это явно и не указано, в антецеденте предполагается достижимость $n^{k_i}_{\alpha_i}(A)$ по A и X по $n^{k_i}_{\alpha_i}(X)$ $(i = \overline{1,s})$ и эти ПВ удовлетворяют исходной схеме S наряду с прочими посылками правила. Как и прежде, предположим, что заключение правила Φ не удовлетворяет схеме S. Рассмотрим два кортежа s и t из типа $S|_{\mathbf{I}}$, которые совпадают по компонентам $A|_{\mathbf{I}}$. В силу предположения s и t должны различаться хотя бы по одной компоненте множества $X|_{\mathbf{I}}$. Пусть для s и t одновременно имеет смысл $\Phi_1|_{\mathbf{I}}$ (удовлетворено условие p), тогда для s и t имеет смысл и $\Phi_1|_{\mathbf{I}}$. Сделанное предположение входит в противоречие с первой посылкой правила, а программный терм в заключении правила вырождается в F_1. Пусть теперь для s и t одновременно имеют смысл ϕ_i $(i = \overline{1,s})$ и $\Phi_2|_{\mathbf{I}}$. Вообще говоря, кортежи (размеченного) типа $S|_{\mathbf{I}}$ могут иметь вид

$$\begin{array}{lll} & true : \langle \dots, A, \dots, X, \dots & \\ & \quad p : \ \dots & \textit{нерекурсивная ветвь} \\ \downarrow & - p : \ \dots & \textit{рекурсивная ветвь} \\ & \quad - p.n^{k_i}_{\alpha_i}(p) : \dots n^{k_i}_{\alpha_i}(A), \dots, n^{k_i}_{\alpha_i}(X), \dots \rangle, & \end{array}$$

здесь подразумевается, что все значимые элементы являются проинтерпретированными.

Мы находимся в рекурсивной ветви $(-p)$, когда кортежи s и t включают фрагмент, помеченный селектором $-p.n^{k_i}_{\alpha_i}(p)$, и *совпадают* по $A|_{\mathbf{I}}$, следовательно, они *совпадают* по $n^{k_i}_{\alpha_i}(A)$ (по правилу суперпозиции и в силу наличия в антецеденте неявных ПВ «перевычисления аргументов»), *совпадают* по $n^{k_i}_{\alpha_i}(X)$ (по правилу суперпозиции и в силу ϕ_i) и, наконец, кортежи обязаны *совпадать* по X (по правилу суперпозиции и в силу наличия неявных ПВ «перевычисления результатов» вида $G \dots n^{k_i}_{\alpha_i}(X) \to X/Q\& - p)$. Таким образом, мы пришли к противоречию с высказанным предположением, и правило (4) исчисления SR также корректно. □

Перейдем теперь к вопросу исследования полноты правил исчисления SR. Для этого нам надо показать, что если некоторое ПВ не может быть выведено из информации о схеме S по правилам SR, то оно не удовлетворяет схеме S. Введем понятие замыкания набора атрибутов относительно схемы S, которое понадобится нам при доказательстве теоремы о полноте исчисления.

Определение 17. Пусть $S(r) = r.(A, X, \cdots \mid filter)$, и пусть A — некоторый набор атрибутов схемы S. *Замыканием* A^+ *для* A *относительно* S будем называть объединение всех таких атрибутов X, что ПВ $F : A \to X$ может быть получено с использованием правил SR из информации о схеме S.

Важное свойство замыкания A^+ определяется следующей леммой.

Лемма 1. ПВ $F : A \to X$ *выводимо по правилам* SR, *если и только если* $X \subseteq A^+$.

Доказательство. Пусть $X = x_1, \dots, x_n$, и пусть $X \subseteq A^+$. Из определения A^+ следует, что для каждого i, $i = \overline{1, n}$, по правилам SR выводимо ПВ $F_i : A \dots, x_i, \dots$, из которого по правилу сужения (0) может быть получено ПВ $F_i : A \to x_i$. Используя схему аксиом $N : A \to A$ и правило суперпозиции, получаем $Fi : A \to A, x_i$. Повторяя эти рассуждения для каждого i, в конце концов, выводим ПВ $F : A \to X$. Обратное, очевидно, следует из определения A^+. □

Сформулируем и докажем теперь теорему о полноте приведенного исчисления.

Теорема 4. *Исчисление* SR *является полным относительно понятия* ПВ, *удовлетворяющего схеме* S.

Доказательство. Пусть S — исходная схема, и пусть ПВ $\Phi \rightleftharpoons F : A \to X$ не может быть выведено на основании информации о схеме S с помощью правил SR. Для доказательства теоремы мы должны построить интерпретацию $\mathbf{I}$ такую, что в $S|_{\mathbf{I}}$ имеются кортежи s и t, для которых $\Phi|_{\mathbf{I}}$ имеет смысл и которые, совпадая по компонентам $A|_{\mathbf{I}}$, различаются хотя бы по одной компоненте из $X|_{\mathbf{I}}$. Рассмотрим интерпретацию $\mathbf{I}$ такую, что в $S|_{\mathbf{I}}$ содержатся два кортежа s и t, компоненты которых совпадают для атрибутов из A^+ и различаются для всех прочих атрибутов схемы S. Предположим противное, т. е. что ПВ $\Phi_1 \rightleftharpoons F1 : B \to Y$ является выводимым, но не удовлетворяет S, и определяется это на зафиксированных кортежах. Из предположения следует, что Φ_1 имеет смысл для s и t и что $B \subseteq A^+$ (в противном случае кортежи не совпали бы по некоторому атрибуту из B). Кроме того, s и t должны различаться по компонентам из Y, т. е. $Y \not\subseteq A^+$ (но, вообще говоря, неверно, что $Y \cap A^+ = \varnothing$). Пусть y — атрибут, $y \in Y \setminus A^+$. Поскольку $B \subseteq A^+$, то ПВ $G : A \to B$ выводимо по лемме 1. Отсюда с учетом Φ_1 по правилу суперпозиции имеем $G; F_1 : A \to Y$, значит, $Y \subseteq A^+$, а сделанное предположение неверно. Итак, ПВ, выведенные с помощью правил SR, удовлетворяют исходной схеме.

Покажем теперь, что Φ не удовлетворяет S, в частности, при интерпретации **I**. Предположим обратное. Поскольку $A \subseteq A^+$, заключаем, что должно выполняться $X \subseteq A^+$, так как иначе s и t, совпадая по компонентам $A|_{\mathbf{I}}$, не совпадали бы по компонентам $X|_{\mathbf{I}}$. Отсюда опять по лемме 1 заключаем, что ПВ $F : A \to X$ может быть выведено по правилам SR. Это противоречит сделанному предположению, а значит, не удовлетворяет схеме S, в частности, на кортежах s и t рассматриваемой интерпретации **I**. Таким образом, мы показали полноту правил SR. $\square$

§ 3. Стратегия и алгоритм установления выводимости предложений SR

Алгоритм, включающий и стратегию вывода, описывается с использованием некоторого псевдокода, структуры управления которого традиционны для императивных языков программирования. При описании алгоритма используются следующие обозначения:

- C_k — множество y-атрибутов (развертки) исходной схемы, достижимость которых установлена на текущий момент;
- *по*-аксиома — *проблемно-ориентированная* или *предметная* аксиома — ФС из текущей схемы;
- «*вход в подсхему*» — наличие в C_k y-атрибутов, которые принадлежат подсхеме текущей схемы;
- A_0 и S берутся из формулировки задачи T.

Описание алгоритма.

«установить $i = 0$, $C_0 = A_0$»;
«поднять флаг “*продолжать доказательство*”» ;
«текущей схемой объявить S»;
while «поднять флаг “*продолжать доказательство*”» **do**
  if «имеется *по*-аксиома текущей схемы, аргументы которой входят в C_i»**then**
    «применить правило суперпозиции и,
      если возможно, правило ветвления»;
    «определить C_{i+1} : к C_k добавить новые y-атрибуты либо изменить
      условия достижимости ранее полученных»;
    «установить $i = i + 1$»;
    «если определился вход в подсхему, запомнить ее в стеке»
  else if «имеется подсхема, в которую определен вход» **then**
      «объявить ее текущей схемой»
  else if «текущая — рекурсивная схема» **then**
      «применить правило введения рекурсии; установить $i = i + 1$»;
      «определить C_{i+1}»
  else «объявить текущей внешнюю схему»
  fi fi fi
  if «текущая — исходная схема S, и больше нет *по*-аксиом,
      аргументы которых входят в C_i» **then**
    «опустить флаг “*продолжать доказательство*”» **fi**
  od.

Для приведенного алгоритма справедлива

Теорема 5. *Приведенный алгоритм корректен, т. е. обеспечивает нахождение всех выводимых атрибутов. Верхней оценкой сложности алгоритма является выражение* $O(|M|^3)$.

Доказательство. (i) Индукцией по i покажем, что если некоторый y-атрибут e помещается в C_i, то $e \in A_0^+$.

База индукции ($i = 0$). $e \in A_0^+$, и по схеме аксиом (0) и правилу (1) получаем $A_0 \to e$.

Индукция. Пусть $i > 0$ и C_i состоит из атрибутов, которые принадлежат A_0^+. Пусть также на $i+1$-м шаге используется ПВ $F : B \to Y$ и $e \in Y$. Имеем $B \subseteq A_0^+$, поскольку $B \subseteq C_i$, следовательно, по лемме 1 есть $G : A_0 \to B$. Используя правила композиции и сужения, получаем $G; F : A_0 \to e$, т. е. $e \in A_0^+$.

(ii) Покажем теперь, что если y-атрибут e входит в A_0^+, то e попадает в некоторое C_i. Точнее, докажем более общее утверждение: если на основании информации о схеме S по правилам SR выведено ПВ $G : A_0 \to B$, то множество y-атрибутов B содержится в некотором C_i.

Вывод ПВ представляет собой последовательность применений правил SR к no-аксиомам и ранее полученным предложениям вычислимости (начальное ПВ $N : A_0 \to A_0$ получается в результате использования схемы аксиом). Доказательство проводим индукцией по длине вывода.

База индукции (минимальная длина вывода).

Вывод заканчивается применением схемы аксиом — A_0 состоит в точности из B, очевидно, $B \subseteq C_0$. Длина вывода два: схема аксиом (0) и правило сужения (1). Здесь, очевидно, также $B \subseteq C_0$.

Индукция. Предположим, что сделанное утверждение истинно для выводов, длина которых не превосходит k, и пусть вывод ПВ $\Phi \rightleftharpoons F : A \to B$ имеет длину $k+1$. Если Φ получается по правилу сужения, то B принадлежит C_k, которое найдется по предположению для вывода, предшествующего получению Φ. Пусть B получено применением правила суперпозиции, например, из полученных ранее $\Phi_1 \rightleftharpoons F1 : A \to Z$ и $\Phi_2 \rightleftharpoons f : Z \to b$ ($B = Z \cup b$). Вывод каждого из двух последних ПВ имеет длину меньше $k+1$. Следовательно, по предположению найдется такое C_i ($i < k+1$), что $Z \subseteq C_i$. По способу введения правила композиции Φ_2 является no-аксиомой — функциональной связью из набора $filter$ исходной схемы S. Отсюда заключаем, что b (а следовательно, и $B = Z \cup b$) попадает в C_{i+j}, где j — число применений правил SR между получением в выводе ПВ Φ_1 до использования no-аксиомы Φ_2 (включительно).

(iii) Правила ветвления и введения рекурсии в контексте доказываемой теоремы можно не различать — здесь нам не важен способ подтверждения предположения, появляющегося в последнем правиле, достаточно просто знать, что мы это делать умеем.

Итак, пусть ПВ $\Phi \rightleftharpoons F : A \to B$ получено по правилу ветвления и длина вывода равна $k+1$. Пусть также Φ получено из $\Phi_1 \rightleftharpoons F_1 : A \to B_1, b/Q\&p$ и $\Phi_2 \rightleftharpoons F_2 : A \to B_1, b/Q\& - p$. Длина вывода как Φ_1, так и Φ_1 меньше $k+1$, следовательно, найдутся C_i и C_j, которые содержат $B_1, b/Q\&p$ и $B1, b/Q\& - p$ соответственно, причем при построении C_i и C_j обсуждаемый алгоритм работает от одних и тех же исходных данных A_0. Отсюда видно, что атрибуты множества B и, в частности, $b \setminus Q$ будут включены в $C_{\max(i,j)+1}$ в силу того, что правило ветвления применяется сразу же, если показана достижимость атрибута при альтернативных условиях. Поскольку по лемме 1 принадлежность атрибута e к замыканию A_0^+ означает выводимость ПВ $F : A_0 \to e$, можно заключить, что обсуждаемый алгоритм корректно строит A_0^+. $\square$

Заключение

Как показывает практика, теория C-моделей вместе с исчислением SR допускает весьма эффективные реализации разнообразных когнитивных систем. В силу того, что задача установления выводимости тесно связана с нахождением транзитивных замыканий специальных отношений, естественные варианты реализации опираются на прекрасно зарекомендовавшие себя и весьма эффективные подходы. Прежде всего среди последних имеет смысл выделить алгоритм Армстронга [7] и алгоритм Диковского [8]. В разных модификациях эти алгоритмы обладают полиномиальными оценками с невысокими степенями (линейные либо квадратичные), хотя и связаны с существенно более бедными по сравнению с описываемыми здесь моделями. В первом случае речь идет о связях на атрибутах реляционных схем в базах данных, а во втором рассматриваются линейные вычислительные модели. Известно, что наиболее распространенная система непроцедурного (логического) программирования [9] базируется на фрагменте классической логики предикатов, определяемом формализмом так называемых хорновских дизъюнктов. Не менее известным является факт неразрешимости классической логики первого порядка. Средства борьбы с неполнотой базовой теории в существующих версиях системы Пролог определяются внелогическими механизмами, подобными известной операции усечения (cut), и не всегда внятными стратегиями вывода. Возможность создания в рамках прикладной теории (спецификации Пролога) предикатов высших порядков еще более усугубляет ситуацию неразрешимости, связанную с классическими исчислениями.

Предлагаемая здесь теория, наследуя свойство полноты исчисления высказываний, является выразительно достаточно мощной, чтобы использоваться для моделирования содержательно богатых предметных областей.

ЛИТЕРАТУРА

1. *Кахро М. И., Каля А. П., Тыугу Э. Х.* Инструментальная система программирования ЕС ЭВМ (ПРИЗ). М.: Финансы и статистика, 1981.
2. *Минц Г. Е.* Резолютивные исчисления для неклассических логик // 9-й Советский кибернетический симпозиум. М.: ВИНИТИ, 1981. Т. 2. С. 34–36.
3. *Минц Г. Е., Тыугу Э. Х.* Полнота правил структурного синтеза // Докл. АН СССР. 1982. Т. 265, № 6. С. 41–60.
4. *Новосельцев В. Б.* Синтез рекурсивных программ в системе СПОРА. Л., 1985. (Препринт / ИТА АН СССР «Алгоритмы небесной механики», № 43).
5. *Дал У., Дейкстра Э., Хоар К.* Структурное программирование. М.: Мир, 1975.
6. *Дейкстра Э.* Дисциплина программирования. М.: Мир, 1987.
7. *Ульман Дж.* Основы систем баз данных. М.: Финансы и статистика, 1983.
8. *Диковский А. Я.* Детерминированные вычислительные модели // Техническая кибернетика. 1984. Т. 84, № 5. С. 84–105.
9. *Робинсон Дж.* Машино-ориентированная логика, основанная на принципе резолюции. М.: Мир, 1970. (Кибернетический сб.; № 7).

Статья поступила 18 сентября 2006 г.

Новосельцев Виталий Борисович
Томский политехнический университет, ул. Ленина, 30, Томск 634050
vbn@osu.cctpu.edu.ru