УДК 004.89

ЭФФЕКТИВНЫЙ НЕРЕЗОЛЮТИВНЫЙ ВЫВОД ДЛЯ ОГРАНИЧЕННОГО ИСЧИСЛЕНИЯ ХОРНОВСКИХ ДИЗЪЮНКТОВ

В.Б. Новосельцев

Томский политехнический университет
E-mail: Vitalii_Novoseltsev@tpu.ru

Представлен механизм, обеспечивающий логический вывод для систем логического программирования типа Пролог. Предлагаемый подход отличается от стандартного метода резолюций и опирается на обратную стратегию в стиле С.Ю. Маслова. За счет введения строго очерченных ограничений на семантику входного языка удается достичь полноты и эффективной (полиномиальной) разрешимости используемой формальной теории.

Введение

Значительная часть современных программных комплексов в той или иной степени подразумевает использование логического подхода: системы поддержки принятия решения, распознавания естественных языков, быстрого прототипирования программ – все то, что объединяется понятием «*Системы, опирающиеся на знания*» (*Knowledge Based Systems*). Одним из основных путей повышения эффективности труда программистов является увеличение «интеллектуальной мощности» среды разработки. Наиболее популярными инструментами в технологических комплексах являются визуализация и агрегация: интерфейсные элементы, заготовки («объекты»), стандартные схемы («связи объектов») и т. д. Однако необходимо согласиться с тем, что все существующие оболочки, так или иначе, подразумевают употребление лингвистических конструкций. Программист, в конечном итоге, вносит явные текстовые указания в соответствующие позиции шаблонных форм. Объектно-ориентированные среды предоставляют пользователю массу возможностей модификации «всего, что можно», однако последний из этой массы выбирает порядка десяти процентов освоенных вариантов, оставляя для прочих позиций назначения *по умолчанию* (см., например, [1]). Анализ сгенерированного распространенными системами программного текста показывает, что около 50 % содержимого является неиспользуемым (избыточным). Популярные в настоящее время *CASE*-системы (*CASE – Computer Aided Software Engineering*) опираются, по большей части, на некоторую трактовку широко известного трансформационного подхода [2] (или его обедненного клона *fold-unfold* [3]), весьма успешно развивавшегося в свое время школой академика А.П. Ершова.

При всей значимости трансформационного подхода, его следует считать достаточно мощным, но не более чем техническим инструментом (недостатки которого отмечены выше), а не фундаментальным базисом востребованных в настоящее время систем быстрого прототипирования (*rapid prototyping systems*). Наиболее обещающим средством реализации программных проектов, по всей видимости, остается логический подход[1].

Оценивая свойства и особенности построения/использования существующих систем программирования, следует вспомнить эссе Г.С. Цейтина (ЛГУ) начала 80-х гг. прошлого века, сравнивающие математический и программистский стили мышления и его же более позднюю работу [4], а также важные публикации Н.Н. Непейводы (см., например, [5]) о математических основаниях программирования.

В статье отстаивается (вероятный, хотя и небесспорный) тезис о том, что в качестве теоретического базиса обсуждаемого инструментария в современных условиях (с учетом масштабов кода и сложности реализуемого управления) могут служить лишь конструктивные варианты полных исчислений. Главным соображением «небесспорности» является сравнение выразительных возможностей языков исчислений высказываний и вариантов предикатных формализмов – «победитель» в этом сравнении очевиден.

Формальная теория

В этой связи представляется весьма заманчивым построить теории, «почти» обеспечивающие выразительность языков первого порядка, но остающиеся полными и полиномиально-разрешимыми. В качестве одного из вариантов достижения этой цели предлагается теория *структурных* (С-) моделей и связанное с ней исчисление ***SR*** [6]. Язык теории С-моделей задает некоторый вариант типизированных формализмов, допускающий рекурсивные определения (в стиле [7]). При этом исчисление ***SR*** позволяет эффективно (не более чем с третьей степенью от длины исходного описания) устанавливать принадлежность объекта тому или иному классу конкретного денотата С-модели и с той же степенью эффективности генерировать алгоритм установления этой принадлежности.

[1] Термин «логический» здесь используется в самом широком значении и, возможно, должен быть заменен на «метаматематический» – Пролог-системы и вычислительные модели, λ-инструментарий, семантические и потоковые сети и т. д. – все это включается в сферу внимания.

Напомним необходимые определения, следуя [6] – к этой же работе мы отсылаем за недостающими здесь формулировками.

Определение 1. Четверку $\Sigma=(A,F,P,D)$, где A, F, P и 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 $E\subset D$ имен примитивных или первичных схем.

Базовой конструкцией теории является *схема отношения*, выступающая некоторым аналогом классического предиката.

Определение 2. Схема (отношения) S объекта r определяется выражением вида

$$S(r) = (r)(S_{01}(a_{01}),..., S_{0N0}(a_{0N0}),$$
$$if\ p_1 \Rightarrow S_{11}(a_{11}),..., S_{1N1}(a_{1N1})\ \square...\square \quad (1)$$
$$p_k \Rightarrow S_{k1}(a_{k1}),..., S_{kNk}(a_{kNk})\ fi\,|\,filter),$$

где $S\in D\backslash E$, $r\in A$. Для всех возможных значений индексов i, j, $S_{ij}\in D$ – *собственные подсхемы* схемы S, $a_{i\text{-}j}\in A$ – ее *собственные величины*, $p_i\in P$ – выбирающие *селекторы*. В правой части (1) r называется *префиксом* схемы, участок до вертикальной черты – *заголовком*, фрагмент *if..fi* – *вариантной* его *частью*, а остальная часть заголовка – *постоянной частью*. Иероглиф *filter* скрывает список уточняемых ниже *собственных* функциональных связей схемы S. Префикс r может «проноситься» в скобочные фрагменты, так что $(r).(S(a),...)\rightleftharpoons(r.S(a),...)\rightleftharpoons(S(r.a),...)$. Аналогично префиксируются имена селекторов, отображений и атрибутов, используемых в конструкциях из набора *filter*. Атрибуты вариантной части считаются «инкапсулированными», таким образом «видимыми извне» параметрами $S(r)$ являются $r.a_{01},...,r.a_{0N0}$.

Определение 3. Выражение вида

$$f{:}a_1,...,a_n\rightarrow a_0, \quad (2)$$

где a_i, $i=\overline{1,n}$ – имена величин, называется *функциональной связью* (*ФС*). В записи (2) f – это имя ФС[2], a_i ($i=1,n$) – аргументы, a_0 – результат ФС.

При задании интерпретации структура (1) фиксирует некоторое подмножество размеченного декартова произведения [8], определяемое ФС из набора *filter*. Подразумевается, что в подмножество попадают только те кортежи (наборы) атрибутов схемы, значения которых удовлетворяют всем ФС этой схемы[3]. – Неформально, мы можем ставить два типа вопросов: (*i*) удовлетворяет ли набор *конкретных значений атрибутов* схеме, и (*ii*) какие наборы удовлетворяют схеме, если зафиксированы значения *некоторых* из атрибутов[4]. Принимаются явные соглашения, связанные с использованием *предопределенных* отношений (порядка и т. д.) для первичных схем. – Так, учет фундаментального для натуральной арифметики отношения $next(n)\rightleftharpoons n+1$, считается не только допустимым, но и необходимым.

Интерпретация спецификации языка Пролог

Предлагаемая дисциплина заключается в следующем:

1. Иерархия типов строится в виде совокупности объявлений в форме, близкой к фиксации башни функционалов, когда каждый элемент явно дефинируется предками:

$$S_0(r_0)\leftarrow(r_0).(S_{01}(r_{01}),...,S_{0k}(r_{0k}));$$
$$(r_0).S_{01}(r_{01})\leftarrow(r_0.r_{01}).(S_{11}(r_{11}),...);...;\ (r_0).S_{0k}(r_{0k})\leftarrow...$$

При этом полагают, что в терминальных конструкциях все $S_{ij}\in E$ («построение от базиса»); считается также, что в записи $S(r)$, *синтаксически* r представляет собой место индивидной подстановки, а *семантически* форма сообщает, что конкретизация r удовлетворяет отношению S (набор *filter* индуцирует *функцию, принадлежности* [9])[5].

2. Реализуется объявление спецификаций в стиле нотации дизъюнктов Хорна: $S(r){:}\!-(r)S_1(r_1),...,(r)S_i(r_i),...$ (синтаксис *почти* аналогичен (1)). В рамках теории С-моделей проблема сопоставления индивидных имен при установлении логических связей решается *явным образом* – прямым указанием точечно-определенного имени в необходимой позиции, а не процедурой унификации в стиле Дж. Робинсона [10]. Префикс r в правой части дизъюнкта является опциональным и может опускаться.

3. Принадлежность интерпретированного кортежа отношению (типу) определяется реализацией на нем всей совокупности отображений из всех множеств *filter* (см. выше разъяснение к Определению 3).

В рамках предлагаемого подхода трактовка и реализация построений, подобных спецификациям Пролога, отличается от классической. Остановимся на этих отличиях подробнее и попытаемся одновременно подчеркнуть преимущества предлагаемой технологии.

Как известно, спецификация на стандартном языке Пролог включает: (*i*) факты в форме предикатов с *константными* параметрами, (*ii*) предложения или клаузы импликативного вида, связывающие предикаты, параметрами которых могут быть пропозициональные *переменные* и (*iii*) целевое

[2] Считается, что реализация f – вычислимая функция.

[3] Можно договориться о том, что любая схема с постоянной частью арности N является k-разрешимой ($0<k<N$), т. е. задание значений любого количества k атрибутов (подчеркнем, постоянной части схемы) уточняет непустое подмножество атрибутов постоянной же части в процедурной форме $\{a_{[k]}\}\rightarrow\{a_{[N-k]}\}$ – модель работы с равенством.

[4] Связь с классическими кванторами «∃» и «∀».

[5] Достаточно очевидно, что символ «←», в соответствующей позиции, является синонимом для «:–».

утверждение, определяемое предикатом, не все параметры которого являются индивидными константами. Важно отметить, что применение для спецификации программы логического языка даже первого порядка неизбежно приводит нас к проблеме, связанной с результатом Гёделя. На практике это вызывает необходимость искусственного торможения резолютивного вывода (управления последним) с помощью внелогических операций, подобных операции усечения («*cut*» или «!») в языке Пролог. Очевидная эклектичность вызывает не менее очевидное недоверие к инструменту[6].

Предлагается следующая трактовка стандартной клаузы языка Пролог (интерпретация фактов и целевого утверждения остается стандартной). Пусть имеет место

$$S(r){:-}\ S_1(r_1),...,S_M(r_M), Q_1(t_1), Q_N(t_N), \qquad (3)$$

где $S_i(r_i)$, $i=\overline{1,M}$, удовлетворяют (1), т. е. выступают «аналогами» классических предикатов[7], $Q_j(t_j)$, $j=\overline{1,N}$ имеют форму ФС (см. сноску 6), причем для любого $j=\overline{1,N}$, $t_j \subseteq \cup r_i$ ($i=\overline{1,M}$)[8]. Приведенные соглашения практически полностью покрывают синтаксис любой стандартной Пролог-системы. Естественно считать, что форма (3) порождает описание вида (1) в соответствие со следующим правилом: для $i=\overline{1,M}$

$$\&(r).S_i(r_i) \rightleftharpoons S(r)=(r).(S_1(r_1),..., S_M(r_M) \mid filter), \qquad (4)$$

где *filter* представляет собой объединение $Q_j(t_j)$, $j=\overline{1,N}$ для всех $i=\overline{1,M}$. Таким образом, мы установили соответствие хорновского предложения и схемы С-модели. Достаточно очевидно, что сделанные предположения несколько сужают общую (робинсовскую) процедуру вывода для исчисления предикатов первого порядка. С другой стороны, анализ почти любой спецификации языка Пролог убеждает в оправданности подобных ограничений. Более того, известно, что трудоемкость алгоритма резолюций (в силу общности последнего) является достаточно высокой (в пределе – экспоненциальной), даже если спецификация соответствует разрешимой проблеме.

Таким образом, в работе продемонстрирована возможность сведения стандартных предложений языка Пролог к конструкциям языка теории С-моделей. В [6] показана полнота этой теории, приведена схема алгоритма логического вывода, а также установлено, что сложность такого алгоритма не превосходит $O(|N|^3)$, где N – константа, соответствующая длине исходной спецификации. Более того, используемый алгоритм реализует обратный вывод в стиле С.Ю. Маслова (см. [10], Приложение 1). В современных программных системах с элементами искусственного интеллекта обязательной составляющей является подсистема объяснения полученного вывода, а отсутствие в алгоритме бэк-трекинга (как в рассматриваемом случае) делает процесс объяснения более простым и естественным по сравнению с традиционным подходом.

Заметим, что форма (3) дает дополнительную возможность введения рекурсии (когда в определении S некоторое $S_i \doteq S$), при этом, правда, можно ожидать конфликта рекурсивного определения S_i вида (1) и, собственно, формы (3). Во избежание подобных ситуаций можно потребовать, чтобы рекурсивность в форме (3) допускалась только для S_i, определение которых вида (1) не является само рекурсивным. С другой стороны, достаточно очевидным является алгоритм выяснения синтаксического «невыхода» из рекурсии при наличии (1), (2) с учетом (3) – см. [6].

Заключение

Инструментарий с элементами искусственного интеллекта на базе пропозициональных исчислений и связанные с ним программные комплексы, весьма вероятно, смогут заполнить нишу в ряду востребованных и надежных технологических оболочек. Более того, комплексы проектирования и программирования, опирающиеся на предлагаемые механизмы, обладая свойством полноты, обеспечат эффективную реализацию сколь угодно сложных прикладных проектов.

[6] Известны примеры спецификаций для (чистого) Пролога, которые не могут быть реализованы машиной вывода Дж. Робинсона. А то, что включаемые в спецификацию Пролога отношения равенства таковыми не являются, обладая направленностью связывания имени и терма, упоминалось уже многократно.

[7] С учетом сделанных выше замечаний, когда r_i отображают только элементы постоянных частей.

[8] Заметим, что в рамках представленной теории символы «$\leftarrow$» и «$\rightarrow$» не выступают взаимно-обратными, а являются независимыми. Аналогично, близкими по смыслу, но семантически неэквивалентными являются знаки «$\rightarrow$», «$\Rightarrow$» и «$\supset$».

СПИСОК ЛИТЕРАТУРЫ

1. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на C++. – М.: Бином; СПб.: Невский диалект, 2001. – 560 с.
2. Касьянов В.Н., Поттосин И.В. Системы конкретизации: подход и основные понятия // Препринт ВЦ СО АН СССР, № 349. – Новосибирск, 1982. – 22 с.
3. Burstall R.M., Darlington J. A Transformation system for developing Recursive Programs. // Journal of the ACM. – 1977. – V. 24. – P. 44–67.
4. Цейтин Г.С. Является ли математика частью информатики? // Компьютерные инструменты в образовании. – 1999. – № 5. – С. 3–7.
5. Непейвода Н.Н. Математик и прикладник: о взаимо(не)понимании // Труды Удмуртского государственного университета. – 2005. – № 7. – С. 23–31.
6. Новосельцев В.Б. Теория структурных функциональных моделей // Сибирский математический журнал. – 2006. – Т. 47. – № 5. – С. 1014–1030.
7. Тейз А., Грибомон П. и др. Логический подход к искусственному интеллекту. Т. 1. От классической логики к логическому программированию. – М.: Мир, 1990. – 322 с.
8. Вирт Н. Структурное программирование. – М.: Мир, 1975. – 189 с.
9. Клини С. Математическая логика. – М.: Мир, 1973. – 480 с.
10. Чень Ч., Ли Р. Математическая логика и автоматическое доказательство теорем. – М.: Наука, 1983. – 360 с.

Поступила 14.04.2008 г.

Ключевые слова:

Логический вывод, полнота теории, пропозициональное исчисление, разрешающий алгоритм, метод резолюций, обратный метод.

УДК 004.89

МЕХАНИЗМЫ ВЫДЕЛЕНИЯ КОНТРАСТНЫХ ФРАГМЕНТОВ СЦЕНЫ СИСТЕМОЙ ВОСХОДЯЩЕГО ВНИМАНИЯ

С.В. Аксёнов

Томский политехнический университет
E-mail: asv@osu.cctpu.edu.ru

Система восходящего внимания анализирует фрагменты зрительных сцен, выделяя наиболее контрастные и информативные области. В работе предлагается подход, расширяющий модель восходящего внимания Л. Итти с целью учета более эффективного расстояния между цветовыми компонентами и снижения влияния к трансформациям и ширине локальных характеристик сцены.

Когда человеческий глаз рассматривает окружающую среду, далеко не все её компоненты воспринимаются одинаково. Некоторые фрагменты автоматически выделяются из своего окружения, и зрительная система настраивается на их интерпретацию. В этом случае говорят, что эти области были локализованы фокусом внимания. Подчеркнем, что данный процесс идет не параллельно, а предшествует распознаванию образов.

В самом общем случае выделение фрагментов в сцене происходит в результате взаимодействия двух потоков внимания. Первый, называемый восходящим, представляет распространение информации от низших отделов зрительной системы (рецепторы) к высшим (корковые структуры). Его задача состоит в выявлении наиболее контрастных областей сцены. Второй, или нисходящий, поток анализа протекает в обратном направлении (от высших до низших корковых отделов) и он служит для локализации фрагментов, синтезируемых источником. В работе рассматривается модель, описывающая работу первого потока, выявляющего наиболее информативные и отличающиеся от общего фона области, попавшие на зрительную сцену.

Основные принципы функционирования современного понимания системы восходящего внимания были заложены К. Кохом и С. Ульманом [1]. Согласно их представлению, весь объем зрительной информации анализируется на определенных цветовых представлениях сцены. Выявление значимых характеристик фрагментов сцены (как, например, цвет или интенсивность) производится двумерными областями нейронов, настроенными на определенный сигнал. Эти области также называются картами признаков. После нахождения активностей всех карт, результаты всех вычислений комбинируются в одном двумерном массиве клеток, называемом картой особенностей. Расположение наиболее активных клеток карты особенностей, найденное согласно процедуре WTA (Winner-Takes-All), позволяет локализовать наиболее информативные области. Другими словами, положение наиболее активной клетки указывает на фрагмент сцены, который попадет в высшие области (центральное представление) для последующей обработки в первую очередь. Далее, выделятся области сцены, ассоциированные со вторым по активности нейроном карты особенностей и т. д. Таким образом, фокус внимания будет переходить от наиболее контрастных сегментов сцены к менее «интересным». Модель такого взаимодействия дана на рис. 1.