ISBN 978-5-901795-16-3 ПРОГРАММНЫЕ СИСТЕМЫ: ТЕОРИЯ И ПРИЛОЖЕНИЯ. Переславль-Залесский, 2009

УДК 004.89

А. Е. Пинжин

Реализация системы логического вывода на основе структурных функциональных моделей для ряда логических исчислений

Аннотация. Рассматривается подход к приведению Хорновских правил, выраженных на языках ряда логических исчислений (логики высказываний, логики первого порядка и дескриптивной логики) к конструкциям теории структурных функциональных моделей. Применение предлагаемого способа интерпретации позволяет использовать алгоритм вывода на структурых функциональных моделях для доказательства ряда логических теорем в перечисленных логиках. Представлены результаты экспериментального сравнения производительности с существующими алгоритмами.

1. Введение

В настоящее время наблюдается значительный интерес к методам интеллектуальной обработки информации. Среди этих методов весомое место занимают алгоритмы, обеспечивающие дедуктивное доказательство логических теорем на основе заданного набора высказваний. Несмотря на длительную историю развития и накопленный опыт в этой области, эффективность алгоритмов логического вывода остается актуальной проблемой. Подход, представленный в [1], предлагает альтернативный алгоритм вывода на основе теории структурных функциональных моделей (С-моделей) [2]. Первоначально этот алгоритм разрабатывался с целью синтеза структурных программ. В [3] показано, каким образом функциональные связи С-модели могут быть интерпретированы в виде дизъюнктов Хорна. Основной целью настоящей статьи является обоснование возможности трансформации задач вывода, сформулированных на базе С-моделей, в логические теоремы классической логики высказываний (ЛВ), логики первого порядка (ЛПП), а также дескриптивной логики (ДЛ). Представлено общее описание реализации машины вывода на С-моделях и результаты опытного сравнения производительности с некоторыми существующими алгоритмами вывода на перечисленных логиках.

2. Основные определения

Исходные данные для машины вывода на С-модели поставляются в виде набора схем. Для упрощения рассуждений будем рассматривать простые схемы с подсхемами без вариантной части и рекурсивных вхождений (более подробная информация представлена в [2]).

Определение 1. *Структурной функциональной моделью* называется конечная совокупность схем вида $M = (T_1, \ldots, T_s)$.

Определение 2. *Простой схемой* T объекта t будем называть выражение вида

$$T(t) = (T_0(a_0), T_1(a_1), \ldots, T_n(a_n)|fset), \tag{1}$$

где T – имя схемы, $a_0, a_1, \ldots, a_n$ – собственные атрибуты (величины) схемы T; $T_0, T_1, \ldots, T_n$ – собственные подсхемы схемы T, определяющие тип соответствующих им атрибутов.

Внутренние атрибуты схемы T, принадлежащие атрибуту $t : T$, выражаются записью $t.a_0, \ldots, t.a_n$. Каждый тип атрибута, встречающийся в определениях схем С-модели M, должен являться примитивным (например, целое число, строка), либо соответствовать схеме, принадлежащей M. Выражение $fset$ в завершающей части описания схемы скрывает множество функциональных связей схемы T.

Определение 3. *Функциональная связь (ФС)* – это выражение вида $f : a_1, \ldots, a_n \to a_0$, где f – имя, $a_1, \ldots, a_n$ – аргументы, a_0 – результат ФС.

Необходимым ограничением для ФС является то, что атрибуты $a_0, a_1, \ldots, a_n$ должны являться собственными атрибутами схемы T. Постановка задачи планирования в С-модели M может быть представлена следующим образом:

$$S = (T(t), A_0, X_0), \tag{2}$$

где t – непервичный атрибут схемы $T \in M$, на которой ставится задача, A_0 и X_0 – наборы имён соответственно исходных и искомых атрибутов, принадлежащих T.

3. Интерпретация С-модели

Традиционная интерпретация I С-модели M, содержащей простые схемы, построенные согласно (1), задает:

для каждой элементарной схемы $T^E \in M$ первичный тип T^E_I;

для каждой ФС $f \in T$ – некоторое отображение $f_I : T_{1I} \times \cdots \times T_{nI} \to T_{0I}$.

Согласно методу интерпретации спецификаций, приведенному в [3], иерархия вложенных атрибутов может быть представлена следующим образом:

$$T(t) \leftarrow t.(T_{01}(a_{01}), \ldots, T_{0n}(a_{0n})); \quad t.T_{01}(a_{01}) \leftarrow t.a_{01}.(T_{11}(a_{11}), \ldots, T_{1m}(a_{1m})); \ldots, \tag{3}$$

где все $T_{ij} \in E$. Переход к примитивным типам позволяет задать интерпретации для различных логических исчислений.

3.1. Логика высказываний

Каждой примитивной величине a_{ij} ставится в соответствие примитивное высказывание A_{ij}, а функциональный символ интерпретируется как знак логического следствия. Тогда множество ФС схемы может быть представлено в виде следующего выражения:

$$A_0 \leftarrow A_{01} \wedge \cdots \wedge A_{0n}, \ldots, A_k \leftarrow A_{k1} \wedge \cdots \wedge A_{kn},$$

т.е. в виде набора Хорновских дизъюнктов:

$$A_0 \vee \neg A_{01} \vee \ldots \vee \neg A_{0n}, \ldots, A_k \vee \neg A_{k1} \vee \ldots \vee \neg A_{kn}. \tag{4}$$

Задача планирования (2), при $A_0 = \{a_1, \ldots, a_i\}$, $X_0 = \{a_{i+1}, \ldots, a_j\}$, для постановки задачи проверки выполнимости набора высказываний (sat-problem) интерпретируется как $X_0 \leftarrow A_0$ и, совместно с (4), соответствует:

$$A_1, \ldots, A_i, \neg A_{i+1}, \ldots, \neg A_j. \tag{5}$$

Приведенная интерпретация позволяет использовать алгоритмы вывода на С-модели [1] для автоматического доказательства весьма обширного множества теорем.

3.2. Логика первого порядка

Во многих классических работах, например [4], описывается метод приведения вывода на логике первого порядка (ЛПП) к выводу в логике высказываний. Приведем некоторый набор высказываний ЛПП, находящихся в стандартной форме и сформулируем задачу вывода:

$$\forall z(A_0(z) \leftarrow A_{01}(z) \wedge \cdots \wedge A_{0n}(z), \ldots, A_k(z) \leftarrow A_{k1}(z) \wedge \cdots \wedge A_{kn}(z), A_1(z), \ldots, A_i(z), \neg A_{i+1}(z), \ldots, \neg A_j(z)). \tag{6}$$

Вполне очевидно, что пропозиционализация этих высказываний приводит теорему к виду (4), (5), а значит и к (2).

3.3. Дескриптивная логика

Интерпретация задачи вывода для дескриптивной логики (ДЛ) представляет интерес, в частности, в связи с интенсивным развитием парадигмы Semantic Web, где дескриптивная логика находит широкое применение. В ряде источников, например [5], приводится сопоставление логики первого порядка и дескриптивной логики (ДЛ). Известно, что выражения ДЛ могут быть приведены к высказываниям ЛППП, если они не вовлекают в предикаты более двух переменных. В таблице 1 приведены некоторые базовые правила трансформации.

Таблица 1. Соответствие выражений ДЛ и ЛППП

ДЛ	ЛППП
$\top$	T
$\bot$	F
R	$R(x)$
$R \subseteq S$	$\forall x(R(x) \leftarrow S(x))$
$R \cup S$	$\forall x(R(x) \vee S(x))$
$R \cap S$	$\forall x(R(x) \wedge S(x))$

Хорновские правила в ДЛ записываются в виде аксиом вида:

$$A_0 \subseteq A_{01} \cap \cdots \cap A_{0n},$$

что эквивалентно следующему выражению ЛППП:

$$A_0(x) \leftarrow A_{01}(x) \wedge \cdots \wedge A_{0n}(x).$$

Задача категоризации в ДЛ: $A_0 \subseteq A_{ij}-?$ аналогичным образом трансформируется в выражение ЛППП. Все это в совокупности позволяет сформулировать следующую постановку проблемы (6) в виде правил ДЛ:

$$A_0 \subseteq A_{01} \cap \cdots \cap A_{0n}, \ldots, Ak \subseteq Ak1 \cap \cdots \cap Akn,$$
$$A_1(z) \equiv \top, \ldots, A_i(z) \equiv \top, A_{i+1}(z) \equiv \bot, \ldots, A_j(z) \equiv \bot. \tag{7}$$

4. Экспериментальные результаты

Реализация машины вывода на С-модели подробно описана в [1]. Приведем здесь лишь краткое описание основных модулей программы (рис. 1).

Для целей сопоставления производительности были разработаны конвертеры формата данных С-модели в различные форматы [6–9], используемые машинами вывода, выбранными для опытного сравнения. В таблице 2 приведен список реализаций машин вывода, вид логики и формат исходных данных.

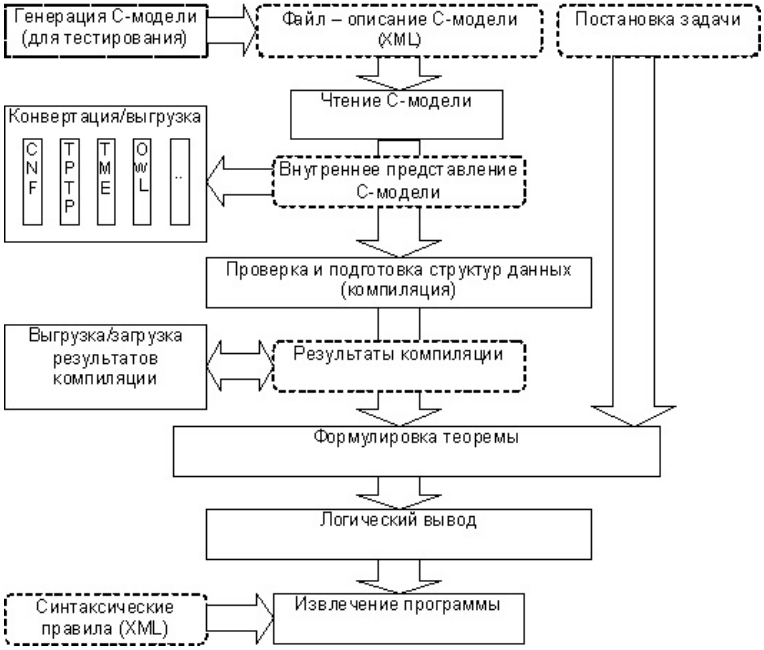


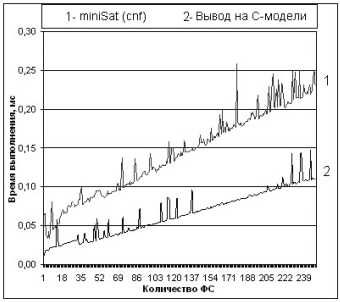
Рис. 1. Структура машины вывода на С-модели

Поясним выбор машин вывода. Пакет Sat4j содержит реализацию алгоритма miniSat, показавшего максимальную эффективность согласно итогам конкурса по решению SAT-проблемы в 2006 г. [10]. Darwin, Paradox, iProver являются номинантами конкурса CADE ATP System Competition (CASC) 2007, проводимого в рамках ежегодной конференции по автоматическому логическому выводу [11]. Отметим, что Vampire 8.0/9.1 не участвует по причине закрытого описания входных параметров. Pellet считается одной из наиболее эффективных свободно распространяемых машин вывода на ДЛ [12].

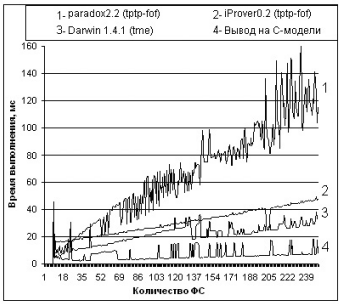
Таблица 2. Форматы данных машин вывода

Машина вывода	Логическое исчисление	Формат данных
miniSat (sat4j)	ЛВ	cnf
Darwin 1.4.1	ЛППП	tme
paradox2.2	ЛППП	tptp-fof
iProver0.2	ЛППП	tptp-fof
Pellet	ДЛ	owl-rdf

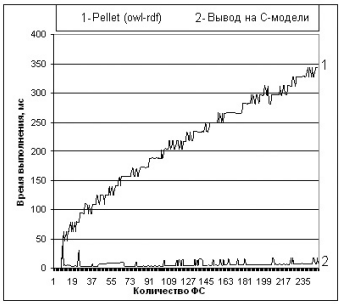
Для целей сравнения выполнялась генерация и конвертация схем с возрастающим от 10 до 250 числом ФС. Аргументы и цели ФС связывались таким образом, чтобы обеспечить необходимость обработки всех элементов при осуществлении вывода. Результаты испытаний приведены на рис. 2.



а)



б)



в)

Рис. 2. Сравнение производительности машин вывода (а – ЛВ, б – ЛППП, в – ДЛ)

Отметим, что при измерении времени работы алгоритмов на ЛПП и ДЛ учитывается время загрузки исходных данных (согласно требованиям CASC [11]), поэтому на рис. 2(а) результаты работы алгоритма вывода на С-модели отличаются от аналогичных результатов рис. 2(б) и рис. 2(в).

5. Заключение

Безусловно, приведенные результаты испытаний нельзя считать точным показателем относительной производительности алгоритмов. Основной целью проведения тестов являлось обоснование того, что предлагаемый алгоритм может быть использован для вывода в различных логиках, показывает сопоставимые показатели эффективности и способен составить конкуренцию существующим реализациям.

Особо следует отметить, что в расчет издержек не включались затраты на представление исходной модели в виде специальных структур данных (компиляция схем), на которых основан алгоритм вывода на С-моделях. Эти структуры подготавливаются однократно и могут быть использованы для постановки любых задач, допустимых в данной модели, однако возможность внесения изменений в подготовленные структуры данных остается предметом дальнейших исследований. Из этого следует, что на текущий момент практическое применение предлагаемого алгоритма структурного вывода обосновано и эффективно для достаточно устойчивых моделей.

Список литературы

[1] Новосельцев В. Б., Пинжин А. Е. *Реализация эффективного алгоритма синтеза линейных функциональных программ* // Известия Томского политехнического университета. — Т. **312**, № 5, 2008, с. 32–35. ↑1, 3.1, 4

[2] Новосельцев В. Б. *Теория структурных функциональных моделей* // Сибирский математический журнал. — Т. **47**, № 5, 2006, с. 1014–1030. ↑1, 2

[3] Новосельцев В. Б. *Эффективный нерезолютивный вывод для ограниченного исчисления хорновских дизъюнктов* // Известия Томского политехнического университета. — Т. **312**, № 5, 2008, с. 94–97. ↑1, 3

[4] Рассел С., Норвиг П. Искусственный интеллект: современный подход (AIMA). — 2-е изд.: Вильямс, 2007. — 381–384 с. ↑3.2

[5] Baader F., Calvaneze D., McGuiness D. L., Nardi D., Patel-Schneider P. F. The Description Logic Handbook: Theory, Implementation, and Applications: Cambridge University Press, 2003. — 154–156 с. ↑3.3

[6] SAT DIMACS Challenge – Satisfiability: Suggested Format: Электронный ресурс, 1993, Режим доступа: ftp://dimacs.rutgers.edu/pub/challenge/satisfiability/doc/satformat.tex. ↑4

[7] TME input specification: Электронный ресурс, Режим доступа: http://www.uni-koblenz.de/ag-ki/Systems/Protein/tme-syntax.txt. ↑

[8] TPTP syntax v3.5.0: Электронный ресурс, Режим доступа: http://www.cs.miami.edu/~tptp/TPTP/SyntaxBNF.html. ↑

[9] OWL Web Ontology LanguageGuide: Электронный ресурс, 2004, Режим доступа: http://www.w3.org/TR/owl-guide/. ↑4

[10] SAT-Race 2006: Runtime comparison of all SAT-Race solvers: Электронный ресурс, 2006, Режим доступа: http://fmv.jku.at/sat-race-2006/analysis.html. ↑4

[11] The CADE ATP System Competition, The 21st International Conference on Automated Deduction: Электронный ресурс. — Bremen, Germany, 17, Режим доступа: http://www.cs.miami.edu/~tptp/CASC/21/. ↑4, 4

[12] Sirin E., Parsia B., Grau B. C., Kalyanpur A., Katz Y. *Pellet: A Practical OWL-DL Reasoner*, № CS 4766. — University of Maryland, College Park, USA, 2005. ↑4

Томский Политехнический Университет

A. E. Pinzhin. *Realization of a reasoner based on structural functional models for several types of logical calculus* // Proceedings of Program Systems institute scientific conference “Program systems: Theory and applications”. — Pereslavl-Zalesskij, v. **1**, 2009. — p. 145–152. — ISBN 978-5-901795-16-3 (*in Russian*).

Abstract. Presented an approach for transformation of Horn clauses, expressed in several logical calculus (propositional logic, first-order logic and descriptive logic) into structural functional models theory constructions. Proposed interpretation method allows to use inference algorithm based on structural functional models for proving logical theorems expressed in the logics listed above. Experimental results of performance comparison with existing algorithms are shown.