УДК 004.42: 004.82:372.8
DOI 10.23951/1609-624X-2017-12-184-187

ПОСТРОЕНИЕ ОБУЧАЮЩЕГО СРЕДСТВА (НА ОСНОВЕ АЛГОРИТМА ПРОВЕРКИ ПРОТИВОРЕЧИВОСТИ МНОЖЕСТВА ДИЗЪЮНКТОВ)

А. Н. Стась, Д. В. Карташов

Томский государственный педагогический университет, Томск

Рассмотрена реализация алгоритма проверки множества дизъюнктов. Используется автоматная грамматика для описания языка представления дизъюнктов, метод резолюции для проверки их противоречивости и поиск в глубину для автоматизации стратегии OL-опровержения. Данный алгоритм может применяться при автоматической проверке доказуемости или недоказуемости теоремы на основе множества некоторых аксиом. Пошаговая детализация данного алгоритма может быть использована в качестве дополнительного средства при обучении методу резолюции и поиску в пространстве состояний, а также основам формальных языков.

Ключевые слова: *дизъюнкт, метод резолюции, OL-опровержение, автоматная грамматика, поиск в глубину, обучение логическим моделям представления знаний.*

Одним из актуальных направлений развития прикладной информатики является использование алгоритмов, основанных на моделях представления знаний. Часто такие алгоритмы называют интеллектуальными. Принято выделять четыре основных класса таких моделей – логические, продукционные, фреймовые и сетевые модели [1]. В процессе обучения будущих IT-специалистов, а также преподавателей и учителей информатики необходимо уделить достаточное внимание различным моделям представления знаний.

Как правило, использование тех или иных моделей обусловлено решаемыми задачами [2, 3]. С этой точки зрения очень широкое применение находят логические модели. В частности, данные модели широко используются при создании экспертных систем и автоматизации доказательства теорем.

Непосредственно для автоматизации логического вывода широко используется метод резолюции. Фактически метод резолюции – это общий подход, позволяющий проверять, является ли какая-то теорема логическим следствием множества аксиом на основе второй теоремы дедукции [4, 5]. За счет того что логические формулы, описывающие аксиомы и отрицание теоремы, можно представить в конъюнктивной нормальной форме, задача фактически сводится к проверке противоречивости множества дизъюнктов (т. е. противоречивости их конъюнкции).

Приведем алгоритм резолюции в общем виде.

ВХОД: множество дизъюнктов С.

ВЫХОД: 1 – если С противоречиво, 0 – в противном случае.

M:=S; // – M – текущее множество дизъюнктов.

while $\square \notin$ M do

Choose (M, c_1, c_2, p_1, p_2) // – выбор родительских дизъюнктов.

if c1, c2 = $\varnothing$ then

return 0 // – нет вариантов для построения резольвенты.

end;

if c:=R(M, c_1, c_2, p_1, p_2); // – вычисление резольвенты.

M:=M $\cup$ {c}; // – пополнение текущего множества.

end;

return 1; //успех

К сожалению, этот метод имеет значительный недостаток, а именно высокую временную трудоемкость (экспоненциальная). В этих условиях актуальным является поиск возможностей сокращения перебора, основанный на использовании той или иной стратегии. Наиболее широкое распространение получили хорновская стратегия (на ее основе фактически создан язык Пролог) и стратегия упорядоченного линейного вывода (OL-вывода) [6, 7]. В качестве средства алгоритмической реализации требуемой стратегии может быть использован поиск в пространстве состояний.

Несмотря на то что хорновская стратегия широко используется, она не лишена недостатков, существенным из которых является неполнота, т. е. стратегия не гарантирует получения пустого дизъюнкта на противоречивом множестве. Зато стратегия OL-вывода обладает полнотой при выполнении незначительного дополнительного условия «Если множество дизъюнктов S противоречиво, а множество {S/C} – выполнимо, где C – некоторый дизъюнкт, то существует OL-опровержение с верхним дизъюнктом C».

Основная идея OL-опровержения состоит в следующем. Будем считать, что литеры в дизъюнкте упорядочены. Для того чтобы сохранять информацию об отрезанных литерах, они не окончательно выбрасываются, а специальным образом помечаются, визуально это можно сделать с помощью обрамления. Если на каком-то шаге выводится дизъюнкт, содержащий две одинаковые литеры (включая символ обрамления), то оставляется самая левая из них, затем отбрасываются все обрамленные литеры, справа от которых нет необрамленных. Эта операция называется отождествлением влево. Если в упорядоченном дизъюнкте последняя литера унифицируется с отрицанием одной из обрамленных литер, то дизъюнкт называется редуцируемым и производится его редукция, т. е. удаляется последняя литера и отбрасываются все обрамленные литеры, за которыми не следуют необрамленные. OL-опровержение предполагает, что отрезаемая литера всегда последняя. Схема OL-опровержения представлена на рисунке.

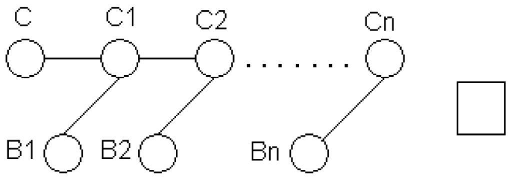

Схема OL-опровержения

Здесь, C1, … , Cn – центральные дизъюнкты, а B1, … , Bn – боковые. Боковой дизъюнкт всегда выбирается либо из первоначального множества опровергаемых дизъюнктов, либо в перечне дизъюнктов, полученных на предыдущих шагах.

Для того чтобы реализовать алгоритм OL-опровержения множества клозов, необходимо решить следующие задачи:

– формализовать язык представления входного множества дизъюнктов с помощью аппарата формальных грамматик;

– реализовать алгоритм OL-опровержения с помощью выбранной стратегии обхода пространства состояний.

Для случая логики высказываний первая задача решается относительно просто. В качестве допустимых обозначений высказываний будем использовать малые буквы латинского алфавита. В качестве обозначения операции дизъюнкции – «,», в качестве знака отрицания – «–».

В этом случае формат представления дизъюнкта можно описать с помощью формальной грамматики, содержащей следующие правила:

S->A|A,S
A->B|-B
B->a|..|z

Нетерминал S соответствует описанию дизъюнкта. Нетерминал A – литере, нетерминал B – атому. Легко видеть, что данная грамматика является контекстно-свободной.

Преобразуем данную грамматику к автоматной. Первую группу правил (для S) получаем за счет замены группы правил S->A, A->B|-B, B->a|..|z на правило S->a|..|z и ввода нетерминала C->,S. Вторую группу правил (для A) получаем заменой правил A->B, B->a|…|z на A->a|…|z.

S->a|...|z|-B|AC|
A->a|...|z|-B
B->a|..|z
C->,S

Группу правил для S получаем за счет замены группы правил S->AC, A->a|..|z|-B на S->aC|…|zC|-BC.

S->a|...|z|-B|aC|...|zC|-BC
B->a|..|z
C->,S

Вводим нетерминал D, равнозначный сочетанию BC.

S->a|...|z|-B|aC|...|zC|-D
B->a|..|z
C->,S
D->BC

Избавляясь от нетерминала B в правиле D->BC, получаем атоматную грамматику, в которую добавляем условный конечный нетерминал Z.

S->aZ|...|zZ|-B|aC|...|zC|-D
B->aZ|..|zZ
C->,S
D->aC|...|zC

Для избавления от недетерминированности вводим нетерминал E={C,Z}, F={B,D}. Проведя замену и избавившись от ставших после замены недостижимыми нетерминалов, получим:

S->aE|...|zE|-F
E->,S
F->aE|...|zE

При этом нетерминал E также фактически будет играть роль конечного.

Тогда данной грамматике соответствует следующая таблица конечного автомата:

	a..z	–	,
S	E	F	⊥
E	⊥	⊥	S
F	E	⊥	⊥

В качестве внутреннего формата представления упорядоченных дизъюнктов удобно использовать последовательности символов литер со связанными с ними признаками наличия или неналичия отрицания и наличия или неналичия обрамленности.

Для реализации перебора методом OL-опровержения применяем обход пространства состояний. Используем следующий алгоритм:

G:={S}; // – пространство состояний G состоит из одной вершины

OPEN:={S}; // – список вершин, которые требуется раскрыть.

CLOSED:=∅; // – список уже раскрытых вершин.

while true do begin

if Null (OPEN) then return (NO); // – если список OPEN пуст, то ответ отрицательный

n:=First (OPEN); // – берем за n первую вершину в списке OPEN.

OPEN:= Tail (OPEN); // – список OPEN без первой вершины.

ADD (n, CLOSED); // – добавляем вершину n в список CLOSED.

if Term (n) then return (YES,);.

M:=раскрыть (n); // – получаем в M список вершин, получившихся при раскрытии вершины n.

ADD (G, n, m); // – добавляем дуги (n, m) в пространство состояний.

if (not (m∈OPEN)) and (not (m∈CLOSED)) then ADD(m, OPEN)

OPEN:=Reorder (OPEN); // – переупорядочиваем список OPEN

end.

Тип обхода пространства состояний зависит от порядка расположения вершин в списке OPEN. При поиске в глубину первой к раскрытию выбирается вершина, наиболее удаленная от начальной. В этом случае можно упростить реализацию данного метода, применив рекурсивную подпрограмму.

Данный алгоритм легко адаптируем для пошаговой детализации. На этой основе авторами разработана обучающая программа-тренажер, позволяющая наглядно продемонстрировать работу алгоритма опровержения множества дизъюнктов методом резолюции. Применение обучающей программы будет способствовать развитию алгоритмического мышления [8–10]. Данное обучающее средство может быть использовано в процессе обучения студентов по дисциплинам «Основы искусственного интеллекта», «Интеллектуальные системы и технологии», «Представление знаний в информационных системах» в рамках направлений подготовки бакалавров «Информационные системы и технологии», «Педагогическое образование (информатика в образовании)».

Список литературы

1. Семенов Н. А. Интеллектуальные информационные системы: учеб. пособие. Тверь: ТГТУ, 2009. 124 с.
2. Избачков Ю. С., Петров В. Н. Информационные системы: учебник для вузов. 2-е изд. СПб.: Питер, 2008. 655 с.
3. Клишин А. П., Стась А. Н., Газизов Т. Т., Горюнов В. А., Кияницын А. В., Бутаков А. Н., Мытник А. А. Основные направления информатизации деятельности Томского государственного педагогоического университета // Вестн. Томского гос. пед. ун-та (TSPU Bulletin). 2015. Вып. 3 (156). С. 110–118.
4. Просолупов Е. В. Курс лекций по дискретной математике: учебное пособие. Ч. 2. Математическая логика. СПб.: СПбГУ, 2013. 74 с.
5. Игошин В. И. Математическая логика и теория алгоритмов: учеб. пособие для вузов. 3-е изд., стереотип. М.: Академия, 2008. 446 с.
6. Достоверный и правдоподобный вывод в интеллектуальных системах / В. Н. Вагин, М. В. Фомина, Е. Ю. Головина, А. А. Загорянская. М.: Физматлит, 2008. 712 с.
7. Гринченков Д. В., Потоцкий С. И. Математическая логика и теория алгоритмов для программистов: учеб. пособие для вузов. М.: КНОРУС, 2010. 206 с.
8. Якименко О. В., Стась А. Н. Применение обучающих программ-тренажеров в обучении программированию // Вестн. Томского гос. пед. ун-та (TSPU Bulletin). 2009. Вып. 1 (79). С. 54–56.
9. Стась А. Н., Долганова Н. Ф. Развитие алгоритмического мышления в процессе обучения будущих учителей информатики // Вестн. Томского гос. пед. ун-та (TSPU Bulletin). 2012. Вып. 7 (122). С. 241–244.
10. Стась А. Н., Прусских О. Н. Формирование алгоритмического мышления в процессе обучения теории графов // Вестн. Томского гос. пед. ун-та (TSPU Bulletin). 2012. Вып. 2 (117). С. 166–169.

Стась Андрей Николаевич, кандидат технических наук, заведующий кафедрой информатики, ведущий инженер-программист лаборатории автоматизации управления и компьютеризации, Томский государственный педагогический университет (ул. Киевская, 60, Томск, Россия, 634061). E-mail: stasandr@tspu.edu.ru

Карташов Денис Васильевич, аспирант, Томский государственный педагогический университет (ул. Киевская, 60, Томск, Россия, 634061). E-mail: DeKar@tspu.edu.ru

Материал поступил в редакцию 25.05.2017.

DOI 10.23951/1609-624X-2017-12-184-187

BUILDING A TRAINING TECHNIQUE (BASED ON THE ALGORITHM OF VERIFICATION OF THE INCONSISTENCY OF THE SET OF DISJUNCTORS)

A. N. Stas, D. V. Kartashov

Tomsk State Pedagogical University, Tomsk, Russian Federation

The use of knowledge-based algorithms is one of the most relevant directions. Often these algorithms are referred to as intelligent. It is accepted to distinguish four main classes of such models: logical, productional, frame-based, and network.

As we know, the use of any given models is driven by the tasks in hand. From this perspective, logical models have gained momentum. In particular, these models are often used in the creation of expert systems and the automation of proofs of theorems.

This article discusses how to implement the validation algorithm for multiple disjuncts. An automatic grammar is used to describe the language of presenting the disjuncts, a resolution method to check their inconsistencies, and a depthfirst search to automate the OL-denial strategy. This algorithm can be used to automatically check provability or indefeasibility of theorems based on many axioms. The step-by-step detalization of this algorithm can be used as an additional tool for teaching the resolution method and searching the state space as well as the basics of the formal languages.

The type of going around state space depends on the order of the points in the list OPEN. When searching in the first point's depth to open it you select the point that is most remote from the intentional one. In this case, you can simplify the implementation of this method by applying a recursive subprogram.

Key words: *disjunct, resolution method, OL-denial strategy, finite-state grammar, depthfirst search.*

References

1. Semenov N. A. *Intellektual'nyye informtsionnyye sistemy: uchebnoye posobiye* [Intelligent information systems: tutorial]. Tver': TSTU Publ., 2009. 124 p. (in Russian).
2. Izbachkov Yu. S., Petrov V. N. *Informatsionnyye sistemy: uchebnik dlya vuzov.* 2-e izd. [Information systems: textbook for universities. 2 edition]. Saint Petersburg, Piter Publ., 2008. 655 p. (in Russian).
3. Klishin A. P., Stas A. N., Gazizov T. T., Goryunov V. A., Kiyanitsyn A. V., Butakov A. N., Mytnik A. A. Osnovnye napravleniya informatizatsii deyatel'nosti Tomskogo gosudarstvennogo pedagogoicheskogo universiteta [Main directions for applying information technologies to the automation of TSPU activities]. *Vestnik Tomskogo gosudarstvennogo pedagogicheskogo universiteta – TSPU Bulletin*, 2015, vol. 3 (156), pp. 110–118 (in Russian).
4. Prosolupov E. V. *Kurs lektsiy po diskretnoy matematike: uchebnoye posobiye. Ch. 2. Matematicheskaya logika* [A course of lectures on discrete mathematics: tutorial. Part 2. Mathematical logic]. Saint Petersburg, SpbSU Publ., 2013. 74 p. (in Russian).
5. Igoshin V. I. *Matematicheskaya logika i teoriya algoritmov: uchebnoye posobiye dlya vuzov.* 3-e izd., stereotip. [Mathematical Logic and theory of Algorithms: textbook for higher schools. 3 edition, stereotype]. Moscow, Akademiya Publ., 2008. 446 p. (in Russian).
6. Vagin V. N., Fomina M. V., Golovina E. Yu., Zagoryanskaya A. A. *Dostovernyy i pravdopodobnyy vyvod v intellektual'nykh sistemakh* [A reliable and plausible conclusion in intelligent systems]. Moscow, Fizmatlit Publ., 2008. 712 p. (in Russian).
7. Grinchenkov D. V., Pototskiy S. I. *Matematicheskaya logika i teoriya algoritmov dlya programmistov: uchebnoye posobiye dlya vuzov* [Mathematical logic and algorithm theory for programmers: textbook for higher schools]. Moscow, KNORUS Publ., 2010. 206 p. (in Russian).
8. Yakimenko O. V., Stas' A. N. Primeneniye obuchayushchikh programm-trenazherov v obuchenii programmirovaniyu [Use of computer tutors in teaching programming]. *Vestnik Tomskogo gosudarstvennogo pedagogicheskogo universiteta – TSPU Bulletin*, 2009, vol. 1 (79), pp. 54–56 (in Russian).
9. Stas A. N., Dolganova N. F. Razvitiye algoritmicheskogo myshleniya v protsesse obucheniya budushchikh uchiteley informatiki [Algorithmic thinking development when training computer science teachers]. *Vestnik Tomskogo gosudarstvennogo pedagogicheskogo universiteta – TSPU Bulletin*, 2012, vol. 7 (122), pp. 241–244 (in Russian).
10. Stas A. N., Prusskikh O. N. Formirovaniye algoritmicheskogo myshleniya v protsesse obucheniya teorii grafov [Shaping the algorithmic thinking in the process of the education graph theory]. *Vestnik Tomskogo gosudarstvennogo pedagogicheskogo universiteta – TSPU Bulletin,* 2012, vol. 2 (117), pp. 166–169 (in Russian).

Stas A. N., Tomsk State Pedagogical University (ul. Kievskaya, 60, Tomsk, Russian Federation, 634061).
E-mail: stasandr@tspu.edu.ru

Kartashov D. V., Tomsk State Pedagogical University (ul. Kievskaya, 60, Tomsk, Russian Federation, 634061).
E-mail: DeKar@tspu.edu.ru