

звичайний алгоритм уніфікації (алгоритм Робінсона) є неефективним.

Результатом статті є алгоритм уніфікації, який працює на базі генетичних алгоритмів, та сформульовано теорему, яка описує застосовність ГА.

За результатами тестування генетичного алгоритму уніфікації можна зробити висновок, що він працює у 3 рази швидше, ніж звичайний алгоритм уніфікації для класу задач h-термів.

Список літератури

1. Капітонова Ю. В. Основи дискретної математики / Ю. В. Капітонова. – К. : Наук. думка, 2002. – 579 с.
2. Кривий С. Л. Дискретна математика: Вибрані питання / С. Л. Кривий. – К. : Видавничий дім «Києво-Могилянська академія», 2007. – 572 с.
3. Рутковская Д. Нейронные сети, генетические алгоритмы и нечеткие системы / Д. Рутковская, М. Пилиньский, Л. Рутковский // Горячая линия. – Телеком, 2006.
4. Baader F. Unification Theory / Franz Baader, Wayne Snyder // Handbook of Automated Reasoning. – Elsevier Science Publishers, 2001.
5. Luger G. Artificial Intelligence: Structures and Strategies for Complex Problem Solving / George F. Luger. – Addison-Wesley, 2008.

O. Zhezherun, V. Melnichuk

THE USE OF GENETIC ALGORITHMS FOR DECISION OF UNIFICATION TASKS

A task of unification is one of classic task of artificial intelligence, used in the inference systems. The traditional way needs the large volumes of calculations, in general case of exponent scale. Therefore task of searching of unificator by an unconventional way looks actual.

The use of genetic algorithms allows to get an additional method in the decision of problem of unification.

Keywords: logical programming, first-order theory, algorithm of unification, genetic algorithm.

Матеріал надійшов 14.06.2013

УДК 004.89

Афонін А. О., Лялецький О. В.

ОСОБЛИВОСТІ ІНТЕЛЕКТУАЛЬНОЇ ОБРОБКИ ІНФОРМАЦІЇ У СУЧАСНИХ СИСТЕМАХ АВТОМАТИЗАЦІЇ МІРКУВАНЬ

У статті аналізуються сучасні парадигми та стилі інтелектуальної обробки інформації в системах автоматизації міркувань.

Ключові слова: автоматизація міркувань, обробка інформації, машинне доведення, Theorema, Isabelle, Coq, Automath, Lambda-Clam, Mizar, SAD.

Вступ

У наш час посилилась увага до створення нових систем автоматизації міркувань як на базі

інтеграції вже наявних інтелектуальних сервісів, так і за рахунок розвитку підходів до їх створення з урахуванням сучасних досягнень в області інформатики та інформаційних технологій.

Обумовлено це тим, що вирішення задач автоматизації інтелектуальної діяльності людини вимагає вміння комп'ютера проводити обчислення та/або аналітичні перетворення, логічні міркування. Нині вже використовується досить широкий спектр систем автоматизації міркувань у прикладних областях, здатних проводити числові викладки та символічні «обчислення», або дедуктивні побудови у середовищі формальної природної мови. Далі наведено опис деяких особливостей і вимог до методів інтелектуальної обробки інформації, спрямований до організації трирівневої архітектури обробки даних в інтелектуальних системах, головним чином пов'язаних з автоматизацією доведень теорем і пошуку висновку у логіках першого і вищих порядків.

Сучасні парадигми інтелектуальної обробки інформації

Особливості інтелектуальної обробки інформації зручно описувати на базі існуючих парадигм побудови, розвитку та користування системами автоматизації міркувань.

Чисельна парадигма відображає засоби і методи наближеного або точного розв'язку задач чистої або прикладної математики, що базується на виконанні чисельних обчислень і моделей та побудові необхідної послідовності дій над скінченними множинами чисел.

Аналітична парадигма базується на можливості комп'ютерів проводити складні символічні перетворення, робити числові (аналітичні) викладки, будувати графіки функцій, задавати математичні моделі визначених процесів тощо. Вона орієнтована на побудову та використання систем комп'ютерної алгебри і виникла в середині 1960-х рр. як альтернатива обчислювальній парадигмі в зв'язку з появою ЕОМ такої швидкодії, інформаційної ємності і гнучкості, що стало можливим програмування складних інтелектуальних процесів.

Дедуктивна парадигма спирається на декларативний спосіб подання та обробки комп'ютерних знань, який ґрунтується на тому, що існуючі математичні знання представляються у вигляді певних формалізованих текстів (як правило, аксіом, визначень, тверджень, теорем та ін.), а додаткові знання беруться з наявних шляхом деяких умовиводів, що звичайно мають вигляд правил побудови (виводу) нових тверджень з уже встановлених. Системи подання та обробки знань, що базуються на цій парадигмі, отримали назву систем автоматизації міркувань, значна час-

тина яких існує у вигляді систем пошуку логічного виведення і доведення теорем, оскільки саме логіко-математичний підхід виявляється найбільш релевантним та ефективним при рішенні задач на проведення дедуктивних побудов і використовується в багатьох прикладних областях.

Індуктивна парадигма базується у математиці на методах доведення, пов'язаних з перевіркою будь-якої властивості відповідно часткового або повного порядку, а у природознавчих науках на переході від часткових спостережень до справедливих висновків у загальному випадку.

Інтеграційна парадигма «об'єднує» усі вищезазначені парадигми і поділяється на два типи: інтегрування на етапі проектування, тобто ще під час розробки системи передбачається наявність у ній підсистем різного роду, можливості її ієрархічного нарощування та підключення до неї вже наявних комп'ютерних сервісів, а також інтегрування на етапі експлуатації, тобто комбінування в одну систему вже готових комп'ютерних сервісів (особливе зацікавлення до їх розробки викликало широке застосування мережі Internet для передачі необхідних даних та їх інтелектуальної обробки відповідно до існуючих стандартів).

Як приклад використання інтеграції на етапі експлуатації можна навести проекти Open Mechanized Reasoning Systems [2] та OpenMath [3], в рамках яких розроблено мови специфікацій та спілкування відповідно в галузі доведення теорем і символічних обчислень.

Відомими представниками інтеграції на етапі проектування є системи Mizar [10] і Theorema [12]. Але найбільше розвинутим підходом до такої інтеграції декларативного типу є так званий Алгоритм Очевидності (АО), запропонований В. М. Глушковым у [1], він фактично веде до окремої парадигми, яка називається евіденційною парадигмою [4], у рамках якої була реалізована система SAD [11].

Перелічимо основні вимоги до мовних засобів, дедуктивної техніки і методів обробки даних у контексті наведених парадигм.

Вимоги до інтелектуальної обробки інформації

Виходячи з парадигм, наведених вище, можна поставити запитання про одночасне проведення досліджень у напрямку розробки формалізованих мов для запису математичних текстів у найзручнішій для користувача формі, у напрямку еволюційного розвитку поняття машинного кроку дове-

дення, у напрямку створення інформаційного середовища, яке впливає на очевидність машинного кроку доведення, і в напрямку побудови методів взаємодії з інформаційним середовищем та використання інтерфейсних засобів.

Зупинимося докладніше на цих основних положеннях з погляду вимог до них.

Мови. Мови повинні задовольняти таким вимогам:

- мати формальний синтаксис і формальну семантику;
- для одержання замкнених текстів повинні забезпечувати можливість формулювання аксіом теорій, визначень, необхідних тверджень, теорем і доведень;
- їх тезаурус повинен бути відділений від їх граматики, що мусить бути розширюваною; вони повинні бути максимально наближені до природних мов математичних публікацій;
- з метою використання добре розвинутих логічних методів пошуку доведення повинні допускати можливість трансляції математичних текстів, записаних на них, у множину формул тієї чи іншої логіки.

Машинне доведення. Ядро будь-якої системи обробки математичних текстів повинна утворювати так звана «очевиднісна (евіденціальна) процедура», призначена для встановлення як коректності верифікованого кроку, так й істинності твердження, що доводиться в цілому засобами деякої дедуктивної техніки. Природно, що для досягнення цієї мети має бути передбачено різні методи посилення «очевиднісної процедури», зокрема:

- пошук допоміжних тверджень та іншої релевантної інформації;
- перетворення звичайних для людини прийомів доведення у машинні кроки доведення;
- використання методів розв'язку рівнянь, необхідних для вирішення поставленої задачі. Наприклад, використання можливостей систем комп'ютерної алгебри.

Інформаційне середовище. Сучасне бачення проблем автоматизації обробки логіко-математичних знань передбачає побудову і розвиток інформаційного середовища (бази математичних знань), що містить описи властивостей, які змінюються в міру одержання нових знань і, в кінцевому результаті, що впливає на поняття машинної очевидності кроку доведення. Для цього повинні використовуватися вже існуючі засоби представлення, здобуття й обробки математичних знань. Дуже важливим на сучасному етапі є використання онтологічних зв'язків.

Інтерфейс. Інтерфейсні засоби повинні передбачати можливості обміну інформацією між різними системами та активного втручання некваліфікованого користувача в процес пошуку розв'язку поставленої задачі з метою полегшити системі знаходження потрібного рішення.

Для ефективного використання систем автоматизації міркувань також велике значення має вибір стилів обробки вхідних даних: стилю формалізації математичних тверджень, стилю доведення, який використовується, і стилю «гранульованості» доведень, що сприймаються.

Стилі інтелектуальної обробки інформації

На стиль формалізації впливають предметна область, попередні відомості, вид визначень (чи є вони комп'ютеризованими), спосіб міркування (чи є він конструктивним, строго типізованим чи заснованим на тому чи іншому численні) та інші. Очевидно, що на цей стиль впливають насамперед використовувана базова логіка і фундаментальні теорії, що втягнуті у формалізацію.

Існують два основні стилі формалізації: один орієнтований на застосування логік вищих порядків (з теорією типів як основи), а інший – на логіку першого порядку (з теорією множин як основи).

Типи застосовуються в більшості добре відомих систем, як-от: Theorema [12], Isabelle [9], Coq [7], Automath [6], Lambda-Clam [9] та інші. Перевага надається використанню теоретико-типового підходу у випадках звертання до індуктивно-визначуваних областей і рекурсивних визначень. І такий підхід цілком відповідає проведенню формалізації програмістських та інженерних концепцій. Однак цей підхід не є ідеальним для формалізації традиційної математики, хоча в більшості згаданих систем використовуються різноманітні виключно математичні теорії.

Як приклад другого стилю можна пригадати добре відому систему Mizar [10], що використовує логіку першого порядку й теорію множин Тарського-Гротендика. Така орієнтація добре відповідає традиційному способу побудови математики. Існує також більший набір математичних тверджень, перевірених системою Mizar.

Система SAD [11] не адаптована до якої-небудь теорії множин (чи іншої фундаментальної теорії), що дає загальну основу для

формалізації. Замість цього при розв'язанні тієї чи іншої задачі пропонується задати необхідний набір попередніх відомостей, виражаючи базові концепції на спеціальній формальній мові, трансльованій у певний різновид мови першого порядку.

Іншою важливою властивістю систем надання допомоги в автоматизації міркувань є *стиль доведення* (міркувань). Так звані інтерактивні системи (першого типу) найчастіше є тактико-керованими. Це означає, що задане твердження доводиться за допомогою послідовності інструкцій, які вводяться в систему. Ці інструкції (тобто тактики) можуть бути примітивними, тобто такими, що являють собою одноразове застосування правила виведення, чи більш складними, подібними до плану доведення або залучення до обробки розглянутої цілі деякого зовнішнього (відносно системи) прувера. До систем такого типу відносяться Isabelle, Coq, Lambda-Clam і низка інших. Робота з системою такого роду виявляється легкою у випадку, коли вона забезпечує витончений набір могутніх тактик, яких достатньо, щоб розв'язати поставлену задачу.

Інший тип систем орієнтовано на декларативне представлення задачі, що розв'язується. Коли твердження, що доводяться, а також за необхідності, їхні доведення записуються певною формальною мовою, яка повинна бути розширюваною засобами (логічного) структурування тексту. Верифікуюча система такого роду повинна мати здатність до перевірки правильності кожного кроку доведення. Найбільш типовими представниками цього підходу є системи Theorema і SAD. Система Isabelle, розширена засобами Isar [5], також може бути віднесена до таких систем.

Дедуктивна потужність систем допомоги в автоматизації міркувань може змінюватися залежно від вимог, що виставляються користувачем. Тому можна вважати, що назва таких систем коливається між системами верифікації доведень та системами пошуку доведень.

Перші приймають на вхід тільки кроки доведення, що мають вигляд застосувань правил виведення, і, отже, повинні бути деталізовані з точністю до правил виведення. Разом з тим другі, тобто системи пошуку доведень, містять методи пошуку виведення та/або планувальники доведень і намагаються заповнити «відсутні» місця доведення. Системи SAD, Mizar і Theorema є представниками систем такого роду.

Різниця між цими двома типами систем не є надто істотною. Якщо можна побудувати дове-

дення теорем тактико-керованою системою, що використовує проміжне введення цілей і автоматичне закриття підцілей, то така система може розглядатися як керована доведенням. Навпаки, якщо кроки доведення, які сприймаються певною системою, керованою доведенням, містять детальні підказки про способи верифікації, то така система може розглядатися як тактико-керована, для якої підказки відіграють роль тактик.

Дедуктивна потужність систем допомоги в автоматизації міркувань може змінюватися залежно від «гранульованості» доведень, тобто вимог, що виставляються користувачем системи. Тому можна вважати, що назва таких систем коливається між системами верифікації доведень та системами пошуку доведень (хоча цей розподіл є умовним). Перші приймають на вхід тільки кроки доведення, що мають вигляд застосувань правил виведення, і, отже, повинні бути деталізовані з точністю до правил виведення. Mizar є прикладом такої системи, хоча сам набір її правил виведення досить великий. Інші системи пошуку доведень містять методи пошуку виведення та/або планувальники доведень і намагаються заповнити «відсутні» місця доведення. Системи SAD, Mizar і Theorema є представниками систем такого роду. Тактико-керовані системи типу Isabelle та Coq звичайно мають вичерпні (для проведення доведень) набори тактик, так що їхня «дедуктивна потужність» може і не мати принципового значення. Тому вони можуть розглядатися як системи пошуку доведень.

Висновки

Як бачимо, існуючі особливості створення та використання інформаційних технологій та діючих інтелектуальних систем вимагають розвитку підходів до виконання дуже складної дедуктивної і аналітичної (символьної) обробки математичної інформації, для чого на сучасному етапі проектуються і реалізуються системи автоматизації міркувань, включаючи системи автоматизації доведень теорем. Проведений опис особливостей та вимог до обробки в них інформації демонструє, що найбільш вдалою є трирівнева архітектура інтелектуальної обробки інформації, коли перший рівень відповідає за лінгвістичну обробку вхідних даних, другий – за проведення звичайних та евристичних спрощень задач, що розглядаються, на сукупності підзадач, а третій – за автоматичне виконання пошуку логічного виведення та/або чисельних і аналітичних перетворень та/або дій з числами.

Список літератури

1. Глушков В. М. Некоторые проблемы теории автоматов и искусственного интеллекта / В. М. Глушков // Кибернетика. – 1970. – № 2. – С. 3–13.
2. Giunchiglia F. Reasoning theories: Towards an architecture for Open Mechanized Reasoning Systems / F. Giunchiglia, P. Pecchiarri, C. L. Talcott // Technical Report 9409–15. – IRST, Trento, Italy. – 1994.
3. Abbott John. A report on OpenMath: A protocol for the exchange of mathematical information / John Abbott, Angel Diaz, Robert S. Sutton // ACM SIGSAM Bulletin. – Vol. 30, Num. 1. – 1996. – P. 21–24.
4. Lyaletski A. Evidential paradigm: a current state / A. Lyaletski and M. Morokhovets // Programme of the International Conference "Mathematical Challenges of the 21st Century". – University of California, Los Angeles, USA. – 2000.
5. Siek J. Practical theorem proving with Isabelle [Electronic resource] / Isar lecture notes / J. Siek. – Mode of access: <http://www.cs.colorado.edu/~siek/7000/spring07/isabelle-notes.pdf>. – Title from the screen.
6. The Automath checker [Electronic resource]. – Mode of access: <http://www.cs.ru.nl/~freek/aut/>. – Title from the screen.
7. The Coq proof assistant [Electronic resource]. – Mode of access: <http://coq.inria.fr/>. – Title from the screen.
8. The Isabelle generic proof assistant [Electronic resource]. – Mode of access: <http://www.cl.cam.ac.uk/research/hvg/Isabelle/index.html>. – Title from the screen.
9. The LambdaCLAM proof planning system [Electronic resource]. – Mode of access: <http://dream.inf.ed.ac.uk/software/lambda-clam>. – Title from the screen.
10. The Mizar project [Electronic resource]. – Mode of access: <http://mizar.org>. – Title from the screen.
11. The SAD system [Electronic resource]. – Mode of access: <http://nevidal.org/sad.en.html>. – Title from the screen.
12. The Theorema project [Electronic resource]. – Mode of access: <http://www.risc.jku.at/research/theorema/description>. – Title from the screen.

A. Afonin, O. Lyaletskiy

FEATURES OF INTELLIGENT INFORMATION PROCESSING IN MODERN SYSTEMS OF THINKING AUTOMATION

The modern paradigms and styles of intelligent information processing in automated reasoning systems are considered in the paper.

Keywords: thinking processing, information processing, machine proof, Theorema, Isabelle, Coq, Automath, Lambda-Clam, Mizar, SAD.

Матеріал надійшов 16.10.2013

УДК 004.42:510.69

Шкільняк С. С.

ЧИСЛЕННЯ СЕКВЕНЦІЙНОГО ТИПУ ДЛЯ ЧИСТИХ ПЕРШОПОРЯДКОВИХ ЛОГІК КВАЗІАРНИХ ПРЕДИКАТІВ

Для чистих першопорядкових композиційно-номінативних логік часткових однозначних, тотальних неоднозначних та часткових неоднозначних квазіарних предикатів побудовано спеціальні секвенційні числення, які використовують предикати-індикатори наявності значень для змінних. Такі числення пропонувано для логік кванторного та кванторно-екваційного рівнів. Для цих числень доведено теореми коректності й повноти.

Ключові слова: логіка, предикат, логічний наслідок, секвенційне числення.

Ефективним апаратом пошуку виведень є числення генценівського, або секвенційного типу. Такі числення побудовано [1] для компози-

ційно-номінативних логік (КНЛ) однозначних часткових предикатів різних рівнів абстрактності й загальності. Використання в програмуванні