



УДК 681.3: 519.68

**ЕВІДЕНЦІАЛЬНА ПАРАДИГМА ТА
ОБРОБКА КОМП'ЮТЕРНИХ МАТЕМАТИЧНИХ ЗНАНЬ**

З.М. АСЕЛЬДЕРОВ, О.В. ЛЯЛЕЦЬКИЙ, Л.З. ФРОЛОВА

Пропонується сучасне бачення програми «Алгоритму очевидності» (Evidence Algorithm, EA), яке ґрунтується на аналізі поточного стану робіт з інтеграції чисельних викладок, аналітичних перетворень та автоматизації міркувань. Показано, що це бачення EA не тільки відповідає інтеграційним тенденціям, а і дозволяє говорити про евіденціальні парадигми.

ВСТУП

Створення високопродуктивних систем програмування, які отримали назву систем комп'ютерних алгебр та систем автоматизації міркувань і призначені для ефективного вирішення природничо-наукових та технічних задач за допомогою ЕОМ, почали проводитись у першій половині 1960-х р. Це час появи обчислювальних машин такої швидкодії, інформаційної ємності і гнучкості, що стало можливим програмування складних інтелектуальних процесів. До їх числа можна віднести і дослідження з розробки та реалізації програми «Алгоритм очевидності» (Evidence Algorithm, EA), запропонованої академіком В.М. Глушковим наприкінці 1960-х — початку 1970-х рр. [1].

У даній роботі пропонується сучасне бачення алгоритму очевидності, яке ґрунтується на аналізі поточного стану робіт з інтеграції обчислювальної, аналітичної та дедуктивної парадигм. Показано, що сучасний стан дослідження алгоритму очевидності не тільки відповідає інтеграційним тенденціям, а і дозволяє говорити про *евіденціальну парадигму*, що ставить питання про одночасне проведення досліджень у таких напрямках: розробка формалізованих мов для запису математичних текстів у найсприятливішій для користувача формі; еволюційний розвиток поняття машинного кроку доведення; створення інформаційного середовища EA, яке впливає на очевидність машинного кроку доведення; розробка засобів інтерактивної обробки математичних текстів.

Під комп'ютерними математичними службами (КМС) будемо розуміти завершені програмні продукти та комп'ютерні системи, які дають користувачеві можливість проводити числові викладки і/або символічні перетворення, і/або логіко-математичні міркування. На сучасному етапі особливе місце

серед КМС займають системи, побудовані на дедуктивній (логічній) парадигмі, що знайшли своє відображення у вигляді так званих систем автоматизації міркувань (САМ) і системи, побудованої на аналітичній (символьній) парадигмі, тобто так звані системи комп'ютерної алгебри (СКА). Останнім часом всі вони привертають пильну увагу з точки зору як їх теоретичного вивчення, так і промислового використання. Це викликано тим, що, з одного боку, сучасні програмні системи, які використовують аналітичні (символьні) перетворення, є комерційно вигідними КМС. З іншого боку, використання формальних (дедуктивних) методів для розробки і реалізації програмних та апаратних засобів стає надзвичайно важливим як через складність і об'ємність задач, які потребують проведення логічних міркувань, так і через те, що зараз спостерігається перехід від теоретичного вивчення формальних методів до їх промислового використання.

Незважаючи на поширення СКА і САМ, багато прикладних областей залишаються поза сферою існуючих СКА і САМ. Зокрема, сфера застосування СКА може бути значно розширена шляхом надання їй дедуктивних властивостей, оскільки відсутність в СКА необхідних виразних можливостей разом із частим порушенням коректності перетворень, обмежує коло застосування СКА. В свою чергу, САМ, хоч і мають необхідні виразні та дедуктивні можливості, які гарантують коректність перетворень, але не достатньо потужні, що не дозволяє проводити безпосередньо аналітичні перетворення та числові викладки. Все це говорить про обмежене використання СКА і САМ не тільки в математиці, але і, що більш важливо, в інженерних застосуваннях.

В середині 1990-х рр. почали з'являтися дослідження по створенню «гібридних» КМС, які втілюють в собі кращі риси САМ і СКА та суттєво збільшують потужність існуючих комп'ютерних служб. Результатом цих досліджень стала так звана *інтеграційна* парадигма, яка об'єднує різноманітні засоби ефективних обчислень, системи комп'ютерних алгебр та системи автоматизації доведень, що максимально наближає інтеграційну парадигму до ідей, закладених у евіденціальну парадигму. На сучасному етапі досліджень евіденціальна парадигма знайшла своє відображення у вигляді системи автоматизованої дедукції (САД), реалізованої в мові Сі на платформі Linux та розташованої на Web-сайті <http://ea.unicyb.kiev.ua>, що дозволяє реалізувати дистанційний Internet-доступ до САД при розв'язуванні задач обробки математичних текстів.

СУЧАСНІ ПАРАДИГМИ ПОБУДОВИ КМС

Чисельна парадигма відображає засоби і методи наближеного або точного розв'язання задач чистої або прикладної математики, яке базується на побудові скінченної послідовності дій над скінченними множинами чисел. Як правило, її використання починається з побудови математичної (континуальної) моделі. Далі здійснюється перехід від континуальної моделі до її дискретного представлення, що дозволяє побудувати різноманітні алгоритми чисельних обчислень, які потім програмуються для ЕОМ у вигляді пакетів прикладних програм.

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

- наявністю задач, які не можуть бути легко виконані людиною за допомогою ручки та паперу і для яких практично відсутні адекватні представлення їх у вигляді відповідних числових алгоритмів;
- необхідністю символічної обробки великих математичних виразів і проведення громіздких аналітичних викладок;
- зменшенням втрат часу і зусиль при знаходженні розв'язання широкого кола природничо-наукових прикладних задач;
- компактністю запису і наочністю представлення аналітичного розв'язання поставленої задачі у порівнянні з її числовим розв'язком.

До числа найтипівіших прикладів застосування СКА відносяться задачі спрощення алгебраїчних виразів, знаходження коренів рівнянь (систем рівнянь), символічне диференціювання, знаходження визначених і невизначених інтегралів, розв'язання диференціальних та інтегральних рівнянь, побудова графіків функцій тощо.

СКА поділяються на системи загального та спеціального призначення. Найбільш характерними представниками систем першого типу є Mathematica, Reduce, Axiom і Maple, другого – Derive, Gap, Magma і CoCoA. До українських досліджень в області систем комп'ютерної алгебри належать роботи [1, 2 – 5].

Як впливає з написаного вище, принципи побудови СКА добре узгоджуються з емпіричним досвідом людини, який має алгоритмічний характер і набутий людиною в процесі знаходження математичних рішень різноманітних природничо-наукових задач. В той же час у сучасних СКА практично відсутні логічні можливості систем автоматизації міркувань.

Дедуктивна парадигма спирається на декларативний спосіб представлення та обробки комп'ютерних знань, який ґрунтується на тому, що існуючі знання представляються у вигляді певних формалізованих текстів (як правило, аксіом, визначень, тверджень, теорем і т. д.), а нові знання виникають як результат деяких умовиводів, що звичайно мають вигляд правил побудови (виводу) нових тверджень. Системи представлення та обробки знань, обґрунтовані на цій парадигмі, отримали назву систем автоматизації міркувань (САМ), більша частина яких існує у вигляді систем доведення теорем (СДТ), оскільки саме логіко-математичний підхід виявляється найбільш релевантним та ефективним при проведенні логічних побудов і використовується в багатьох прикладних галузях.

Зрозуміло, для того, щоб деяка СДТ могла здійснювати дедуктивну допомогу людині, їй необхідна відповідна мовна «оболонка»; правильно сформульовані аксіоми, гіпотези і правила побудови доведення нових тверджень теорії, вміння генерувати припущення, які доводяться.

У даний момент як лінгвістична оболонка найчастіше використовується мова класичної логіки першого порядку, хоча є можливість вживати мови некласичних логік та логік вищих порядків. Ці мови дозволяють задавати інформацію у вигляді формальних речень, які можуть опрацьовуватися СДТ. Крім того, вони дозволяють уникати появи двозначностей, властивих природним мовам, таким, як російська чи англійська.

Доведення, які породжуються СДТ, дають чітке розуміння того, на основі яких аксіом та гіпотез отримано висновки про справедливість розглянутих припущень. Вони не лише свідчать про їх логічне слідування з аксіом і гіпотез, але й часто описують процеси знаходження необхідних рішень, що має велике прикладне значення. В даний момент СДТ отримують доведення за допомогою спеціальних машинно-орієнтованих методів встановлення істинності припущень, що перевіряються. В них часто вбудовуються і прийоми доведення, які віддзеркалюють життєвий досвід працюючих в різних проблемних галузях експертів. Таким чином, дедуктивна парадигма являє собою певну інформаційну технологію, яка дуже підходить для спроби отримання комбінації точного машинного мислення з природними можливостями людини.

До галузей успішного застосування СДТ відносяться логіка, математика, а також комп'ютерні, інженерні та соціальні науки. Потенційно є багато інших областей, де могли б успішно застосовуватися СДТ. До них належать біологічні науки, медицина, комерція і т. ін.

Існує велика кількість активно використовуваних систем доведення теорем. На рівні логіки першого порядку треба відзначити Vampire, Otter, SETHEO, SPASS і Waldmeister. Серед систем вищих порядків — Coq, HOL, Isabelle та Ngthm.

Інтеграційна парадигма. Можна виділити два типи інтеграції КМС: 1 — інтегрування на етапі проектування, тобто ще під час розробки системи передбачаються наявність у ній КМС різного роду, можливості її ієрархічного нарощування та підключення до неї вже наявних служб; 2 — інтегрування на етапі експлуатації, тобто комбінування в одну систему вже готових КМС (особливе зацікавлення до розробки таких КМС спричинило широке застосування мережі Internet для передачі необхідних даних, що, у свою чергу, викликало створення відповідних стандартів).

Можна вказати цілу низку причин інтеграції різних, в тому числі й різнорідних комп'ютерних математичних служб, поміж яких треба відмітити такі: розв'язання математичної задачі вимагає використання обчислень, аналітичних перетворень, міркувань і доведень; СКА є доволі потужними та гнучкими, але часто результати їх роботи потребують ретельної перевірки, а можуть і взагалі видатися неправильними, тоді як СДТ забезпечують надійність результату, але їм не вистачає спеціалізованих розв'язуючих процедур та евристик, які є у СКА; наявні системи загального призначення (наприклад, Axiom, Maple та ін.) не завжди такі ж ефективні для розв'язання задач, як програми спеціального призначення, орієнтовані на задачі певного класу.

Найбільш відомим представником підходу до інтеграції на етапі проектування є система Mizar, яка в ідейному плані має багато спільних рис із алгоритмом очевидності. Іншим представником може бути проект Theorema.

Відносно інтеграції на етапі експлуатації відмітимо, що в рамках проєктів Open Mechanized Reasoning Systems та OpenMath розроблено мови специфікацій та спілкування відповідно в галузі доведення теорем та символічних обчислень, і ці мови допускають поєднання.

OpenMath — це стандарт передачі математичних об'єктів. Для розробки стандарту створено Європейський Консорціум (European OpenMath Consortium), до якого приєдналася група північноамериканських організацій (Centre for Experimental and Constructive Mathematics, PolyMath Group, IBM, Waterloo Maple Inc.), які сформували об'єднання NAOMI (North American OpenMath Initiative).

Інтегровані системи корисні у багатьох важливих прикладних галузях, таких як пошук доведень у різних математичних теоріях, автоматизація міркувань, верифікація програм та апаратури, перетворення схем програм, пошук логічних висновків з урахуванням певних умов або вимог (так званих constraints), дистанційне навчання і т. ін.

ЕВІДЕНЦІАЛЬНА ПАРАДИГМА

Алгоритм очевидності (Evidence Algorithm, EA) був запропонований у 1970 р. академіком В.М. Глушковым у вигляді програми робіт із створення комп'ютерних систем автоматизації пошуку доведень теорем для надання допомоги математикам в автоматизованій обробці математичних текстів. (В.М. Глушковым у другій половині 60-х рр. також був ініційований аналогічний підхід до створення сім'ї комп'ютерних алгебр у вигляді ієрархії систем, які розвиваються і виконують аналітичні перетворення, серед яких спеціалізовані ЕОМ серії МИР з мовою АНАЛІТИК [2], системи АНАЛІТИК-79 [3] для СМ ЕОМ 1410 та АНАЛІТИК-93 [4], АНАЛІТИК-2000 [5] для ПК типу IBM PC.)

Основні аспекти реалізації проєкту:

- побудова формалізованої мови для запису доведень теорем у різних змістовних галузях математики, причому така мова має бути наближена до природної, яку використовують математики;
- формалізація поняття очевидності у побудованій мові: кожний крок доведення має бути обґрунтованим при використанні формальних методів доведення, а також вже відомих систем математичних фактів;
- побудова та розвиток (за допомогою EA) інформаційної бази, яка містить опис математичних понять та їх властивостей;
- побудова засобів інтерактивної обробки математичних текстів.

Одним з основних у проєкті EA було поняття очевидності кроку доведення, яке змінюється із розвитком EA.

«...Організація алгоритму очевидності має допускати можливість необмеженого поповнення його все новими й новими блоками з метою створення більш високих рівнів ієрархії. Для практичних застосувань алгоритму очевидності важливе значення має досягнення такого рівня його розвитку, при якому середня довжина доведення (включно із побудовою спростовуючих прикладів) зрівняється із середньою довжиною доведень, необхідних у підручниках та монографіях, а згодом — і в спеціальних статтях. При цьо-

му, зрозуміло, крім власне алгоритму очевидності, має розвиватися інформаційна база системи, яка містить опис (мовою практичної математичної логіки) різного роду понять, що використовуються у конкретній математичній теорії, що розглядається. А також властивостей цих понять, процедур побудови та дослідження прикладів і т. п. Все це інформаційне багатство має використовуватися алгоритмом очевидності, подібно тому, як це робиться людиною...»[1].

Ідея створення ЕА відповідає сучасним тенденціям побудови та інтеграції комп'ютерних математичних служб. Термін «комп'ютерна математична служба» використовується для позначення будь-якого програмного продукту і/або комп'ютерної системи, що надає у розпорядження користувача можливості проводити числові викладки і/або символні перетворення, і/або логіко-математичні міркування. Особливістю поточних робіт з ЕА є передбачення того, що пошук і доведення твердження повинні проводитися в середовищі замкнутого математичного тексту, записаного формальною мовою, близькою до природної мови математичних публікацій. (Під замкнутим математичним текстом розуміється текст, який містить усе необхідне для розв'язання задачі пошуку і/або верифікації розглянутого твердження.) У зв'язку з цим основні положення програми ЕА на сучасному етапі можуть бути деталізовані у вигляді так званої **евіденціальної парадигми**.

Мови повинні задовольняти такі вимоги:

- мати формальні синтаксис і семантику;
- для одержання замкнутих текстів забезпечувати можливість формулювання аксіом теорій, визначень, необхідних тверджень, теорем і доведень;
- їх тезаурус має бути відділеним від їх граматики, яка повинна бути такою, що може розширюватися;
- максимально наближатися до природних мов математичних публікацій;
- з метою використання добре розвинутих методів пошуку доведення допускати можливість трансляції математичних текстів, записаних на них, у множину формул мови першого порядку.

Дедукція. Відповідно до евіденціальної парадигми, ядро будь-якої системи обробки математичних текстів повинна утворювати так звана «очевиднісна» (евіденціальна) процедура, призначена для встановлення як коректності верифікованого кроку, так і істинності твердження, що доводиться в цілому в термінах деякої дедуктивної техніки. Природно, що для досягнення цієї мети повинні бути передбачені різноманітні засоби посилення «очевиднісної» процедури, зокрема: пошук допоміжних тверджень та іншої релевантної інформації; застосування аналітичних перетворень (тобто використання можливостей систем комп'ютерної алгебри); перетворення машинних кроків доведення в звичайні для людини прийоми і т.ін. (Розуміється, що всі ці засоби повинні мати можливість обмінюватися даними за допомогою визначеної формалізованої мови (мов).) У зв'язку з цим на сучасному етапі передбачається, що дедуктивна техніка сама по собі повинна:

- відбивати синтаксичну структуру розв'язуваної задачі;
- давати можливість здійснювати ефективні дедуктивні побудови в сигнатурі початкової теорії;

- вміти зводити досягнення поставленої мети до розв'язання ряду допоміжних підзадач;
- мати можливість використовувати найбільш відомі методи автоматичного доведення (резольюції, зворотний метод, цілеорієнтовані генценовські числення і т.п.);
- застосовувати аналітичні перетворення (використовувати можливості сучасних комп'ютерних алгебр), щоб відокремити знаходження рішень від дедуктивного процесу;
- вибирати методи розв'язання рівнянь, необхідні для розв'язання поставленої задачі;
- застосовувати звичайні для людини прийоми доведення (у першу чергу, розкриття визначень і застосування допоміжних тверджень);
- передбачати інтерактивний режим пошуку.

Тому за основу розробки дедуктивної техніки було взято секвенціальний формалізм.

Інформаційне середовище. Програма ЕА передбачає побудову і розвиток інформаційного середовища (бази математичних знань), що містить описи властивостей, які змінюються з одержанням нових знань і, зрештою, впливає на поняття машинної очевидності кроку доведення. Вона має використовувати вже існуючі засоби подання, здобування й обробки математичних знань.

Інтерфейс. Інтерфейсні засоби повинні передбачати можливість активного втручання некваліфікованого користувача у процес пошуку рішення поставленої задачі і полегшувати людині розуміння процесу його пошуку, наприклад, в інтерактивному режимі.

ПОТОЧНИЙ СТАН ЕВДЕНЦІАЛЬНОЇ ПАРАДИГМИ ТА СИСТЕМА САД

Сьогодні можна говорити про першу програмну реалізацію евіденціальної парадигми під назвою «Система автоматизації дедукції» (САД). Зупинимось детальніше на основних властивостях САД, які відрізняють її від інших систем.

Мовні засоби системи САД. Розроблена англійська версія формальної мови під назвою ForTheL (FORmal THEory Language) [6].

З синтаксичної точки зору будь-який ForTheL-текст являє собою набір розділів, які можуть містити фрази та розділи більш низького рівня. Певні розділи, наприклад, теореми, докази і означення, грають таку ж особливу роль в ForTheL-текстах, яка присутня у звичайних математичних текстах.

Мова ForTheL містить мову першого порядку як власну підмножину, що дозволяє представляти різні формули логіки першого порядку у вигляді ForTheL-фраз. Ці фрази містять так звані «концепції» і допускають дієслівне написання мовних одиниць.

Концепція є аналогом терму мови першого порядку, що допускає розглядання параметризованої множини об'єктів як його значення.

Дієслівне написання дає можливість конструювати ForTheL-фрази, використовуючи підмети, предикати, об'єкти та атрибути.

Логічні зв'язки разом з предикатами та концепціями являють собою неподільні одиниці мови ForTheL. Інші синтаксичні одиниці можуть бути введені через означення.

Наведемо приклад замкнутого ForTheL-тексту з теорії множин.

Definition. Let S be a set. A subset of S is a set X such that every element of X is in S .

Definition. Let S be a set. S is empty iff S has no elements.

There exists an empty set.

Proposition. Let S be a set. S is a subset of every set iff S is empty.

Згідно із поточним підходом до пошуку доведення в стилі EA математичний текст, що обробляється, записується як ForTheL-текст. Далі цей текст автоматично трансформується у так званий ForTheL1-текст, який складається з речень, з одного боку, аналогів формул мови першого порядку, а з другого — вони зберігають сигнатуру початкового ForTheL-тексту, його синтаксис і структуру, тобто розбиття на розділи (зазначення, допоміжні твердження та теореми, що доводяться).

Після завершення трансформації ForTheL-тексту в ForTheL1-текст можна побудувати інформаційне (ForTheL1)-середовище для процедур пошуку доведень. Це і стає причиною трансляції в системі САД ForTheL-тексту, який розглядається, у відповідний ForTheL1-текст.

Пошук логічних висновків в системі САД. Як зазначалося раніше, логічні методи САД базуються на секвенціальному формалізмі, що дозволяє конструювати логічні методи, які здійснюють пошук доведень у сигнатурі початкової теорії (наприклад, [7–10]). Цей вибір пояснюється тим, що секвенціальні виведення мають вигляд більш «природних» доведень, ніж виведення, отримані, наприклад, резолюційною технікою. Ця властивість секвенціального формалізму стає важливою, коли передбачається взаємодія людини з комп'ютером.

Для пошуку логічних висновків використовуються імплементації числення GD [9], які перманентно поновлюються.

Вони забезпечують:

- збереження структури початкової задачі;
- редукцію твердження, яке доводиться, до сукупності нових допоміжних тверджень;
- відокремлення обробки рівностей від дедукції з метою керування процесами знаходження розв'язків «рівнянь»;
- створення спеціальної техніки обробки кванторів, яка дозволяє відмовитись від попередньої сколемізації;
- застосування техніки, необхідної для знаходження розв'язків «рівнянь» (тобто застосування звичайної уніфікації, АС-уніфікації, Е-уніфікації і т.п.) та правил обробки рівностей (наприклад, парамодуляції);
- застосування методів доведення, звичайних для людини, наприклад, таких, як зазначення та допоміжні твердження;
- гнучкий інтерактивний режим пошуку логічних висновків.

Інформаційне середовище системи САД. Розрізняються три види інформаційного середовища (ІС) системи САД, які можуть бути використані для представлення й обробки математичних знань.

Зовнішня ІС — це розподілені віртуальні дані, які існують у вигляді сучасних комп'ютерних математичних служб. Доступ до неї може здійснюватися, наприклад, через інтелектуального Internet-агента. Вона використовується для пошуку відповідей на запитання в «зовнішньому» світі на основі вже існуючих засобів добування математичних знань.

Резидентна ІС містить усю сукупність математичних знань про предметні області, у яких працював і/або працює САД. Вона представляє собою сукупність ієрархічно упорядкованих «страт». Страти формуються в процесі еволюційного розвитку САД і повинні використовуватися як для видачі відповідей (з різними рівнями подробиць) на математичні питання із «зовнішнього» світу, так і для створення різних середовищ доведення САД, що відбивають поточні дедуктивні можливості САД щодо розв'язуваної задачі і які можуть бути реалізовані в комп'ютері.

Основна функція *внутрішньої* бази знань — формування і підтримка динамічного середовища доведення для процедур пошуку логічних висновків в системі САД, зокрема, апарата розкриття визначень та пошуку допоміжних тверджень. Інформаційною основою побудови внутрішньої бази знань є резидентна ІС.

Інтерфейсні засоби системи САД. Розробка секвенціального формалізму на базі ForTheL1-модифікації з урахуванням перелічених вище принципів системи САД дозволяє будувати методи пошуку доведення в рамках сигнатури вихідної теорії. Це, у свою чергу, дає можливість генерувати достатні твердження й одержувати наслідки в досить звичному для людини вигляді, що може послужити відправною точкою створення гнучких інтерфейсних засобів.

ОСОБЛИВОСТІ ІМПЛЕМЕНТАЦІЇ СИСТЕМИ САД

Система САД орієнтована на виконання таких процедур:

- синтаксичний аналіз ForTheL-тексту, який готується людиною та містить твердження, яке доводиться;
- компіляція ForTheL-тексту в замкнутий ForTheL1-текст;
- побудова середовища доведення для проведення дедукції;
- пошук доведення;
- редагування доведення з метою отримання математичного тексту у звичайному для людини вигляді.

Імплементована версія системи САД має два базових модуля: транслятор з мови ForTheL в мову ForTheL1 та прuver секвенціального типу для логіки першого порядку.

Транслятор з ForTheL в ForTheL1 розв'язує дві основні задачі: синтаксичний аналіз тексту вхідного ForTheL-тексту із подальшим генеруванням вихідних ForTheL1- даних. При цьому транслятор підтримує словник введених понять, функцій та предикатів.

Транслятор реалізовано для англійської версії ForTheL-граматики, але нині розробляється українська та російська версії мови ForTheL. Це дасть можливість перекладати ForTheL-тексти на різні мови.

Прuver системи САД складається з головної програми, яка має назву «головний ланцюжок перетворень», та декількох допоміжних підпрограм. Головний ланцюжок перетворень складається з трьох кроків: обробки вхідних даних, генерування середовища доведення і пошук доведення теореми.

На першому кроці конструюються усі структури даних, які представляють ForTheL-текст.

На другому — утворюються пари літер, які взаємодіють, оцінюються відповідні підстановки, вибираються шляхи ефективного пошуку доведення та передбачаються ірелевантні дії. Оскільки прuver базується на численних GD [9], замість дерева секвенцій будується дерево літер, що дозволяє значною мірою економити обчислювальні ресурси.

На третьому кроці прuver здійснює пошук доведення, користуючись добре відомим методом пошуку «на початку у глибину». Також передбачено керування процесом пошуку.

У прuverі важливу роль грає модуль обробки рівнянь, який породжує уніфікатор та перевіряє його на допустимість. В ньому також передбачена можливість підключення до зовнішнього «розв'язувача задач».

Для роботи у логіці першого порядку з рівністю реалізована певна модифікація метода Бранда [11].

У найближчому майбутньому передбачається розширити першу версію системи САД до версії, яка впроваджує такий ланцюжок перетворень:

ForTheL-текст з теоремою, яка доводиться

(використовується синтаксичний ForTheL-аналізатор)

Замкнутий ForTheL1-текст

(використовується прuver з розширеними Internet-можливостями)

Машинне ForTheL-доведення виділеної теореми

(використовується редактор ForTheL1-доведення)

Текст з доведенням теореми в звичному для людини вигляді

ВИСНОВКИ

Розглянуто евіденціальну парадигму для автоматизованого пошуку доведень математичних теорем, особливістю якої є те, що пошук доведення здійснюється на базі цілісного математичного тексту, який записано формальною мовою першого порядку, близькою до мови математичних публікацій. Це дає можливість використовувати формальні аналоги таких природних способів доведення, як застосування означень понять та допоміжних тверджень у поєднанні з процедурами пошуку виводу у логічних численнях (а саме, з процедурою, яка базується на машинно-орієнтованому секвенціальному численні). Особливості евіденціальної парадигми створюють можливість для оптимізації перебору при пошуку доведень математичних теорем та використання уже діючих комп'ютерних математичних служб.

Дослідження, що проводяться у даний час з розвитку евіденціальної парадигми, будуть корисними при розв'язанні таких проблем, як розподілене автоматизоване доведення теорем, перевірка коректності математичних текстів, формалізація знань на основі математичних статей, побудова баз знань для математичних теорій.

ЛІТЕРАТУРА

1. Глушков В.М. Некоторые проблемы теории автоматов и искусственного интеллекта // Кибернетика. — 1970. — № 2. — С. 3–13.
2. АНАЛИТИК (алгоритмический язык для описания вычислительных процессов с использованием аналитических преобразований) / В.М. Глушков, В.Г. Боднарчук, Т.А. Гринченко, А.А. Дородницына, В.П. Клименко, А.А. Летичевский, С.Б. Погребинский, А.А. Стогний, Ю.С. Фишман // Кибернетика. — 1971. — № 3. — С. 102–134.
3. АНАЛИТИК-79 / В.М. Глушков, Т.А. Гринченко, А.А. Дородницына, А.М. Драж, В.П. Клименко, С.Б. Погребинский, О.Н. Савчак, Ю.Ф. Фишман, Н.П. Царюк / Ин-т кибернетики АН УССР. — Препр. — Киев, 1983. — № 12. — 73 с.
4. АНАЛИТИК-93 / А.А. Морозов, В.П. Клименко, Ю.С. Фишман, Б.А. Бублик, В.Д. Горовой, Е.А. Калина // Кибернетика и системный анализ. — 1995. — № 5. — С. 127–157.
5. АНАЛИТИК-2000 / А.А. Морозов, В.П. Клименко, Ю.С. Фишман, А.Л. Ляхов, С.В. Кондрашов, Т.Н. Швалюк // Математические машины и системы. — 2001. — №1,2. — С. 66–99.
6. Vershinin K., Paskevich F. ForTheL - the Language of Formal Theories // IJ Information Theories and Applications. — 2000. — 73. — P. 121–127.
7. Degtyarev A., Lyaletski A., Morokhovets M. Evidence Algorithm and Sequent Logical Inference Search // Lecture Notes in Artificial Intelligence. — 1999. — 1705. — P. 44–61.
8. Avidens Algorithm. Linguistic Tools and Deductive Technique of the System for Automated Deduction / Z. Aselderov, K. Verchinin, A. Degtyarev, A. Lyaletski, A. Paskevich // 3rd International Workshop on the Implementation of Logics. Tbilisi, 18-24 October, 2002. — P. 16–30.
9. SAD, a System for Automated Deduction: a Current State / K. Verchinin, A. Degtyarev, A. Lyaletski, A. Paskevich // Proceedings of the Workshop on 35 Years of Automating Mathematics. — Heriot Watt University, Edinburgh, Scotland, 10–13 April, 2002. — P. 12.
10. Degtyarev A., Lyaletski A., Morokhovets M. On the EAStyle Integrated Processing of SelfContained Mathematical Texts // Symbolic Computation and Automated Reasoning (the book devoted to the CALCULEMUS2000 Symposium: edited by M. Kerber and M. Kohlhase), A.K. Peters, Ltd, USA. — 2001. — P. 126–141.
11. Brand D. Proving Theorems with the Modification Method // SIAM Journal of Computing. — 1975. — 4. — P. 412–430.

Надійшла 18.06.2003