

ОСОБЛИВОСТІ ОРГАНІЗАЦІЇ СУЧАСНИХ СИСТЕМ АВТОМАТИЗАЦІЇ МІРКУВАНЬ**Вступ**

Автоматизація доведень теорем (АДТ) – галузь штучного інтелекту, що зародилася невдовзі після впровадження перших ЕОМ: перша програма для пошуку доведень теорем – „Логік-теоретик” – з’явилася у 1956 р. Підходи до розробки засобів та систем АДТ можна розглядати у декількох площинах: з боку організації системи АДТ, тобто складу системи та способів зв’язку між її компонентами, і з боку формалізмів, що лежать в основі процедур пошуку доведень теорем. Останні розділяються на формалізми, які в той чи інший спосіб імітують процес доведення теорем людиною (наприклад, логічні числення секвенціального типу, метод семантичних таблиць, натуральне виведення, доведення за індукцією, доведення за аналогією), та формалізми, що враховують, насамперед, здатність обчислювальної машини швидко виконувати велику кількість однотипних дій і не використовують зразків міркування, притаманних людині (наприклад, матричні методи, зворотний метод Маслова, метод резолюцій Робінсона). Щодо організації відмітимо, що на початковому етапі розвитку системи АДТ будувалися здебільшого за принципом „чорного ящика”. Така система являє собою автономну програму, що реалізує деяку процедуру пошуку доведення теорем, причому можливість впливу на хід пошуку з боку користувача або не передбачається зовсім, або значною мірою обмежена, введення початкових даних та виведення результату пошуку здійснюється за допомогою спеціальної формальної мови (наприклад, мови формул логіки першого порядку, мови диз’юнктив тощо), не передбачається також можливість сполучення такої системи з іншими, спорідненими або допоміжними, системами. В.М. Глушковым був запропонований [1] проект системи для автоматизованого пошуку доведень теорем, відомий під назвою Алгоритм Очевидності (АО), в основі якого лежав відмінний від згаданого вище підхід до організації системи АДТ. Основна ідея підходу: система АДТ має бути помічником працюючого математика, отже має підтримувати весь комплекс робіт, що стосуються доведення теореми (виведення, алгебричні перетворення, обчислення, побудова контрприкладів), а також накопичувати математичні знання і користуватися ними при доведенні теореми, спілкуватися з користувачем мовою хоча і формалізованою, але близькою до природної, надавати користувачеві можливість впливу на хід доведення.

У більшості систем АДТ, створених протягом 1970 – 1980-х рр., використовувався перший тип організації, значна їх кількість ґрунтувалася на принципі резолюцій Робінсона [2].

Досвід експлуатації систем АДТ свідчив, що програми, які створювалися для доведення теорем, виявилися придатними при розв’язанні інших задач, зокрема, побудови моделей та контрприкладів, формулювання та перевірки припущень, дослідження теорій і почали застосовуватися як помічники у міркуванні. З огляду на це цілком природним було виникнення (у 1980 р.) терміна „автоматизоване міркування” (АМ), який почав використовуватися замість терміна „автоматизоване доведення теорем”. Однак досвід роботи виявив недоліки організації системи АДТ як автономної програми, зокрема, при розв’язанні задач, що потребують і доведень, і обчислень, і аналітичних перетворень. Аналогічні проблеми виникли і при використанні систем комп’ютерної алгебри (СКА). В результаті на початку 1990-х рр. з’являються засоби сполучення СКА та АДТ [3]. Поява комп’ютерних мереж і пов’язаних з ними нових способів передачі та обробки даних сприяла формуванню стійкої тенденції до інтеграції різноманітних програмних засобів розв’язання математичних задач (так званих комп’ютерних математичних служб, або КМС) [4]. Інтеграція програм дає можливість, зокрема, розширити коло задач, які розв’язують як системи АДТ, так і СКА, широко використовувати спеціалізовані пакети програм як допоміжні математичні служби, створювати та поповнювати сховища математичних знань з доступом через електронну мережу.

Сучасні системи автоматизації міркувань

На сьогодні існують різні форми і способи інтеграції програмних засобів розв'язання математичних задач, зокрема, автоматизації міркувань. Коротко охарактеризуємо їх.

1. Забезпечення доступу через мережу Internet до засобу АДТ, який створювався початково як автономна програма. При цьому потрібні засоби опису способів звернення до програми і отримання результатів. Прикладом програми АМ, що використовує такий тип інтеграції, є відома програма OTTER [5], яка ґрунтується на методі резолюцій.

2. Інтеграція різноманітних КМС у межах однієї системи. Такий тип інтеграції характерний для Системи автоматизації доведень (САД) [6], реалізованої у межах проекту АО, а також для систем АПС [7], THEOREMA [8], MIZAR [9], ISABELLE [10], Ω MEGA [11].

Оскільки САД в історичному плані є першою системою даного типу інтеграції, наводимо схему її організації (рис. 1). На рисунку К – користувач; TL-текст – математичний текст, записаний формалізованою мовою TL [12], що наближена до природної (TL в САД – мова спілкування користувача з системою, а також мова уведення даних та виведення результатів роботи системи); СС – синтаксична система, що забезпечує перетворення TL-тексту у внутрішнє представлення і навпаки; СДС – секвенціальна дедуктивна система, що здійснює пошук виводу на основі числення α -секвенцій [13]; РДС – резолюційна дедуктивна система, що здійснює пошук виводу на основі резолюційного числення [14]; СД – середовище доведення – інформаційна

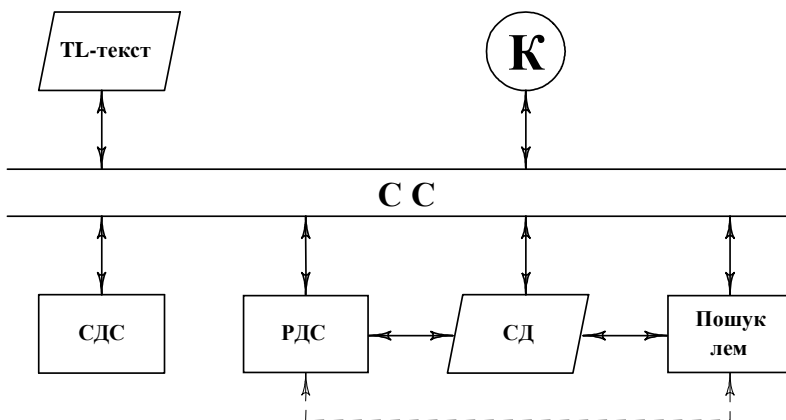


Рис. 1

база для пошуку доведення, що являє собою сукупність визначень та допоміжних тверджень (у внутрішньому представленні) з вхідного TL-тексту, які можуть використовуватися при пошукові виводу; пошук лем – компонент САД, що залежить від ситуації доведення, яка визначається станом простору пошуку виведення, здійснює пошук допоміжних тверджень (лем), застосованих в поточній ситуації доведення (більш детально про

взаємодію цього компонента САД і резолюційної процедури пошуку виводу викладено у [15]). Основою інтеграції дедуктивних засобів САД служить мова TL, а більш точно – внутрішній рівень її представлення в системі. Зауважимо, що зараз створюється версія САД, яка базується на мові ForTheL [16], новій версії TL.

Система алгебричного програмування АПС є засобом розробки прототипів прикладних програм та їх реалізації у середовищі мови Сі. Мова програмування АПС є декларативною. В основі системи лежить загальний механізм переписування термів, що дозволяє використовувати АПС як інструмент для створення різноманітних процедур символічних перетворень, зокрема, дедуктивних засобів [17]. Для реалізації процедури пошуку виводу у середовищі АПС потрібно записати специфікацію цієї процедури у формі логічного числення. У межах АПС можуть співіснувати різні математичні служби. Семантика їх взаємодії базується на теорії агентів та середовищ [18]. Основою інтеграції засобів розв'язання математичних задач в АПС служить механізм переписування термів.

Метою проекту THEOREMA є надання (користувачеві) апарату, що дає змогу виконувати обчислення, розв'язувати задачі та доводити теореми. Реалізація проекту передбачає створення системи, яка включає базу знань K , що складається з сукупності формул та механізму виводу L , механізму обчислення виразу T з K при заданому значенні змінної, яка входить в T , розв'язувача задач для K , що забезпечує знаходження тих

значень змінної (x) , при яких $T(x)$ виконується відносно K , процедуру доведення для K , за допомогою якої можна з'ясувати, чи істинний вираз $T(x)$ для довільного x . Особливостями проекту є: базовим засобом реалізації проекту є система символічних обчислень Mathematica, точніше, та її частина, яка включає мовні засоби; для побудови ієрархії математичних областей використовуються функтори; процедура пошуку доведень підтримує так званий природний тип міркування, тобто імітує стиль міркування, притаманний людині; передбачається наявність низки спеціальних процедур пошуку доведень (відповідно до функторів, що використовуються в системі), а також спеціальних процедур, призначених для пошуку доведення у деяких областях; наявний засіб представлення доведень природною мовою; планується механізм автоматичного породження складних баз знань за алгебричними властивостями даної предметної області та за означенням функтора.

Розробку системи MIZAR розпочато понад 20 років тому з метою створити на базі класичної логіки та теорії множин програмне середовище для підтримки написання математичних статей. MIZAR мав перевіряти правильність математичних текстів, написаних людиною. Зараз до складу системи входять: мова MIZAR, програмний інструментарій, бібліотека, журнал гіперпосилань. Мова MIZAR створювалася як спроба формалізувати мову математичних публікацій. В системі використовується класична логіка. Є можливість за допомогою означень вводити типи, терми, конструктори формул. Інструментарій призначено для розвитку бази даних та керування нею, перевірки логічної коректності текстів, забезпечення узагальнення, спрощення, поліпшення текстового представлення, подання текстів у форматах TEX та HTML. Математична бібліотека системи MIZAR – сукупність текстів (понад 600) на мові MIZAR. Система розвивається як сховище перевірених формальними методами математичних фактів, організованих як система знань. Отже, така система може вважатися математичною службою.

Система Isabelle є засобом інтерактивного доведення теорем. Вона має апарат задання логік. До її складу входить база математичних знань, яка містить бібліотеку сучасних математичних понять.

Ω MEGA є інструментальною системою для доведення теорем, яка підтримує розподілені обчислення за рахунок залучення зовнішніх математичних служб. Особливістю Ω MEGA є наявність планувальника доведення. Процес пошуку доведення в Ω MEGA здійснюється не на рівні конкретного числення, а радше як процес планування доведення на більш абстрактному рівні методів доведення. Планувальник служить в Ω MEGA основою інтеграції. Зараз досліджується можливість побудови планувальника, який здатний користатися власним досвідом пошуку доведень і навчатися новим методам доведень. На базі системи розвивається новий підхід до пошуку доведень теорем, згідно з яким для кожного правила виводу числення, що використовується для пошуку доведення, існує агент (або декілька), який визначає застосовність цього правила виводу на поточному кроці доведення.

3. Інтеграція двох різних систем, які початково створювалися як окремі програмні засоби розв'язання математичних задач. Для забезпечення такого виду інтеграції створюються спеціальні програми-зв'язки, так звані „містки”. Як правило, таким способом сполучаються СКА та системи АМ (САР). В результаті або розширюються можливості програм АМ (наприклад, комбінування СКА Maple з HOL та Isabelle наділило останні потужними засобами арифметичних обчислень і підвищило ефективність цих програм у міркуваннях стосовно поліномів та чисел), або функціонально збагачуються СКА (так, наприклад, САР NUPRL є алгебричним оракулом для СКА WEYL). Діяльність у цій сфері докладно висвітлено в [19].

4. Інтеграція сукупності систем з метою утворення з них „робочої команди”, призначеної для розподіленого розв'язання задач. Цей підхід виник як природне узагальнення попереднього. Він орієнтований на таке використання різноманітних наявних систем обчислень і символічних перетворень, при якому досить гнучко можна формувати потужний обчислювальний комплекс залежно від конкретної задачі, яка підлягає розв'язанню. У межах даного підходу створюються загальні засоби підтримки розподіленого розв'язання задач.

На сьогодні ведуться розробки: а) стандарту для передачі та представлення математичних знань (OpenMath [20]); б) стандарту для опису математичних служб (OMRS [21]); в) інструментальних засобів компонування процедур доведення для формальних міркувань у прикладних галузях (PROSPER [22], MSB [23]); г) архітектур, що забезпечують взаємодію різних КМС [24, 25].

Для забезпечення спілкування між різними системами при розподіленому розв'язанні задач потрібні відповідні мовні засоби. Саме з цією метою ведеться розробка OpenMath – стандарту для передачі математичних об'єктів від однієї програми до іншої. Згідно з цим стандартом, математичні об'єкти представляються за допомогою позначених дерев (OpenMath-виразів або об'єктів), які визначаються рекурсивно через найпростіші OpenMath-вирази (цілі числа, символи, визначення яких знаходяться у так званих змістових словниках, змінні, числа з рухомою комою, рядки символів, масиви байтів) та символи операцій, які використовуються для побудови складених виразів. Це є так званий абстрактний рівень представлення об'єктів. Об'єкти, що призначені для зберігання або передачі, відображаються у послідовності байтів (кодуються). OpenMath забезпечує двійкове кодування та кодування за допомогою XML. Наприклад, OpenMath-вираз $\text{binding}(\lambda x, \text{application}(\sin, x))$ кодується таким чином: `<OMOBJ><OMBIND><OMS cd="fns" name="lambda"/><OMBVAR><OMV name="x"/></OMBVAR><OMA><OMS cd="transc" name="sin"/><OMV name="x"/></OMA></OMBIND> </OMOBJ>`. У змістових словниках зберігаються значення математичних термінів (так званих символів), що стосуються тієї чи іншої предметної області. У кожному словнику є розділ, який містить опис цього словника. Розділ, що стосується символу, містить ім'я символу, його опис (природною мовою), може включати приклади використання символу та його властивості. Сукупність змістових словників поповнюється. Оскільки OpenMath може служити для передачі не тільки математичних виразів, а й таких об'єктів, як доведення, цей засіб може використовуватися для забезпечення спілкування між СКА та САМ.

Іншим засобом інтеграції систем, що активно розвивається зараз, є OMRS (Open Mechanized Reasoning Systems) – апарат, що дає змогу описувати як СКА, так і САМ. Специфікація систем, виконана стандартними засобами, розглядається авторами OMRS як відправна точка інтеграції КМС. OMRS-специфікація має три рівні: рівень логіки, призначений для опису числення (мови та правил виводу), з якими працює система (для опису використовуються спеціальні математичні об'єкти – теорії міркувань); рівень управління, що служить для опису стратегій пошуку виводу (опис здійснюється за допомогою так званих анотованих теорій міркування – об'єктів, що містять інформацію про те, як застосовувати правила виводу); рівень взаємодії, що визначає спосіб взаємодії системи із середовищем (відповідні засоби знаходяться в стадії розробки).

Проект PROSPER (Proof and Specification Assisted Designed Environment) спрямований на розробку інструментарію, що дає змогу експерту порівняно легко та гнучко „збирати” знаряддя для доведення теорем з метою використання у прикладних галузях. Основою для збирання такого знаряддя є HOL, але можна підсилювати його за рахунок підключення зовнішніх інструментів.

MSB (Mathematical Software Bus) – це програмне середовище для сполучення різнорідних систем, які виконують будь-які види математичних обчислень та перетворень. Для підтримки обміну математичними даними пропонується використовувати OpenMath. На основі MSB передбачається проектування інтелектуальних помічників для розподіленого розв'язання математичних задач.

MathWeb являє собою розподілену мережну архітектуру для автоматичного та інтерактивного доведення теорем. Для побудови протоколів спілкування математичних служб використовуються OMRS та KQML (Knowledge Query and Manipulation Language – мова та протокол для обміну інформацією та знаннями між програмними агентами) [26].

LBA (Logic Broker Architecture) – знаряддя, що забезпечує необхідне підґрунтя для співпраці САМ та СКА. Експериментальна реалізація LBA виконана на базі промислового стандарту CORBA [27], що забезпечує взаємодію типу „клієнт-сервер”, та стандарту OpenMath. Складовими LBA є клієнт (САМ, що має потребу у

логічній послугі, тобто звертається із замовленням довести деяке твердження при заданих посилках), сервер (САР, що може надати замовлену послугу), брокер (програма, яка обслуговує клієнта та сервер, передаючи їхні повідомлення один одному, відшукуючи для клієнта той сервер, який може надати потрібну логічну послугу, перекладаючи запит клієнта на мову сервера), база даних (в ній зберігаються відомості про КМС, які можуть виступати в ролі серверів, та про види послуг, які вони пропонують). Дуже важливою складовою ЛВА є модуль перевірки сумісності клієнта та серверу (у розумінні можливості використання результатів, які отримав сервер, для потреб клієнта), але ця складова існує зараз поки що на рівні проекту.

Висновки

Підводячи підсумок, відмітимо, що на відміну від ранніх програм АДТ (але не усіх: згадаймо САД), при створенні яких увага зосереджувалася майже виключно на виборі формалізму пошуку доведення теорем і розробці ефективних стратегій пошуку, сучасні САМ відзначаються порівняно складною організацією. Особливостями організації сучасних САМ є такі:

- відкритість у тому розумінні, що організація САМ і засоби спілкування та взаємодії загального характеру дають можливість нарощувати її за рахунок додаткових засобів символічних перетворень або/та залучати зовнішні математичні служби безпосередньо у процесі розв'язання задачі даною системою, або/та виконувати цій системі роль математичної служби для іншої системи чи при колективному розв'язанні задач;
- наявність розвинутих мовних засобів спілкування з людиною. Роль таких засобів подвійна: по-перше, вони полегшують введення даних в систему, сприймання та подальше використання результатів її роботи, тобто підвищують рівень комфортності праці користувача, а, по-друге, дають змогу людині спостерігати за ходом розв'язання задачі та ефективно впливати на пошук розв'язку. З огляду на те, що ідеологія багатьох систем ґрунтується на трактуванні САМ як помічника людини, яка або навчається математиці, або веде дослідження, або розв'язує складну задачу, мовна складова стає невід'ємною частиною сучасної САМ;
- наявність у складі системи бази математичних знань та її використання під час роботи;
- наявність засобів планування доведення.

Розробка інструментарію для підтримки спілкування та взаємодії систем (таких, як згадані вище стандарти та архітектури), а особливо прогрес в цій галузі, матимуть, на наш погляд, значний вплив на розвиток майбутніх програмних засобів розв'язання математичних задач, зокрема САМ. По-перше, оскільки можна залучити під час роботи підходящу математичну службу, зникає необхідність у зосередженні у межах однієї системи усіх засобів розв'язання задачі. По-друге, підвищується роль окремих спеціалізованих програм, які ефективно реалізують потужні методи розв'язання (нехай і не дуже широкого класу) задач, за умови, що такі програми оформлені як здатні до спілкування та взаємодії математичні служби. Разом з тим зростає значення досліджень із загальних методів пошуку розв'язків задач. Питання методів пошуку вже зараз розглядається на новому рівні. Якщо при реалізації ранніх систем АДТ перевага надавалася певній окремій стратегії пошуку, то в сучасних САМ мова йде про більш гнучкий засіб – механізм планування розв'язку. У майбутніх САМ передбачається втілення механізму вибору методу розв'язання задачі, узагальнення досвіду роботи системи та генерація нових методів розв'язання [28]. Отже, принагідно зауважимо, що зароджується нова тенденція розвитку САМ – створення систем, здатних до навчання.

З викладеного вище випливає, що діапазон сучасних САМ є досить широким і включає ті програми та системи, що мають різні можливості, але, насамперед, здатні до спілкування та взаємодії. Отже, до сучасної САМ можуть бути віднесені:

- математична служба (у вигляді окремої програми), яка ефективно розв'язує задачі в деякій (можливо, досить обмеженій) галузі;
- система загального призначення (тобто орієнтована на розв'язання широкого класу задач), яка є відкритою для спілкування та взаємодії;

- система, ядром якої є компонент побудови міркування, але яка забезпечує увесь цикл розв'язання задачі (можливо, за рахунок співробітництва із зовнішніми математичними службами);
- команда, що складається з підходящих комп'ютерних математичних служб і зібрана спеціально для розв'язання задач певного класу або для розв'язання деякої складної проблеми.

СПИСОК ЛІТЕРАТУРИ

1. Глушков В.М. Некоторые проблемы теории автоматов и искусственного интеллекта // Кибернетика. – 1970. – № 2. – С. 3 – 13.
2. Робинсон Дж. Машинно-ориентированная логика, основанная на принципе резолюции: Киберн. сб. (нов. сер.). – М.: Мир, 1970. – № 7. – С. 194 – 218.
3. Harrison J., Thery L. Reasoning about the Reals: the Marriage of HOL and Maple // Lect. Notes in Artif. Intelligence. – Springer-Verlag, 1993. – Vol.698. – P. 351 – 353.
4. Лялецький О.В., Мороховець М.К. Про деякі тенденції розвитку комп'ютерних математичних служб // Вісник Київського університету. Серія: фіз.-мат. науки. – Вип. 1. – Київ, 1998. – С. 193 – 204.
5. McCune W. Otter 3.0 reference manual and guide // Techn. Report ANL-94-6, Argonne National Laboratory. – Argonne, IL, USA, 1994.
6. Глушков В.М. Система автоматизации доказательств // Автоматизация обработки математических текстов. – Киев: ИК АН УССР, 1980. – С. 3 – 30.
7. Letichevski A.A., Kapitonova Yu.V., Konozenko S.V. Computation in APS // Theoret. Comput. Sci. – 1993. – N119. – P. 145 – 171.
8. A Survey of the Theorema Project / B.Buchberger, T.Jebelean, F.Kriftner et.al. // Intern. Symp. on Symbolic and Algebraic Computation (ISSAC'97), Maui Hawaii, USA. – USA: ACM Press, 1997. – P. 384 – 391.
9. Rudnicki P., Trybulec A. On equivalents of well-foundedness // Journal of Automated Reasoning. – 1999. – N 23 (3–4). – P. 197 – 234. <http://merak.uwb.edu.pl/~bancerec/introduction/>.
10. Paulson L. The Foundation of a Generic Theorem Prover // Journ. of Automated Reasoning. – 1989. – N 5. – P. 363 – 396.
11. ΩMEGA: Towards a mathematical assistant / C.Benzmüller, L. Cheikhrouhou, D. Fehrer et. al. // In: W.McCune, editor, // Proceedings of the 14th Conf. on Automated Deduction, Townsville, Australia. – Springer, 1997. – LNAI. – Vol. 1249. – P. 252 – 255.
12. О формальном языке для записи математических текстов / В.М. Глушков, К.П. Вершинин, Ю.В. Капитонова и др. // Автоматизация поиска доказательств теорем в математике. – Киев: ИК АН УССР, 1974. – С. 3-36.
13. Дегтярев А.И., Лялецкий А.В. Логический вывод в системе автоматизации доказательств // Математические основы систем искусственного интеллекта. – Киев: ИК АН УССР, 1981. – С. 3 – 11.
14. Мороховець М.К. Реализация резолюционных процедур в САД // Дедуктивные построения в системах искусственного интеллекта и моделирование автономных роботов. – Киев: Ин-т кибернетики им. В.М. Глушкова АН УССР, 1987. – С. 25 – 31.
15. Атаян В.В., Мороховець М.К. Сочетание формальных процедур поиска логического вывода и естественных приемов поиска доказательств теорем в системе автоматизации доказательств // Кибернетика и системный анализ. – 1996. – № 3. – С. 151 – 182.
16. Паскевич А.Ю. Особливості реалізації мови високого рівня для запису математичних текстів // Вісник Київського університету. – 1999. – Вип. 2. – С. 284 – 288.
17. Капитонова Ю.В., Летичевский А.А., Волков В.А. Дедуктивные средства системы алгебраического программирования // Кибернетика и системный анализ. – 2000. – № 1. – С. 17 – 34.
18. Letichevski A.A., Gilbert D.R. A general theory of action languages // Cybernetics and Systems Analysis. – 1998. – N 1. – P. 16 – 36.
19. Calmet J., Homann K. Classification of communication and cooperation mechanisms for logical and symbolic computation systems // Frontiers of Combining Systems, Series on Applied Logic. Dordrecht: Kluwer, 1996. – N 3. – P. 221 – 334.
20. Abbott J.A., van Leeuwen A., Strottmann A. *OpenMath*: Communicating Mathematical Information between Co-operating Agents in a Knowledge Network // Journal of Intelligent Systems, 1998. – Special Issue “Improving the Design of Intelligent Systems: Outstanding Problems and Some Methods for their Solution”.
21. Giunchiglia F., Pecchiarri P., Talcott C. Reasoning Theories: Towards an Architecture for Open Mechanized Reasoning Systems // Proc. of the First International Workshop on Frontiers of Combining Systems, Munich. – Germany, 1996.
22. The PROSPER Toolkit / L.A.Dennis, G.Collins, M.Norrish et.al. // Proc. of the International Conference on Tools and Algorithms of Systems. – Berlin, Germany, 2000. – April.
23. Calmet J., Homann K. Towards the Mathematical Software Bus // Theoret. Comput. Sci. – 1997. – N 187. – P. 221 – 230.
24. Armando A., Zini D. Interfacing Computer Algebra and Deduction Systems via the Logic Broker Architecture // Proceedings of the 8th Symposium on the Integration of Symbolic Computation and Mechanized Reasoning. – Scotland, St.Andrews, 2000. – P. 52-67.
25. Armando A., Kohlhasse M., Ranise S. Communication Protocols for Mathematical Services based on KQML and OMRS // Ibid. – P. 37 – 51.
26. Finin T., Fritson R. KQML – a language and protocol for knowledge and information exchange // Proc. of the 13th Intl. Distributed Artificial Intelligence Workshop. – Seattle, WA, USA, 1994. – P. 127 – 136.
27. Object Management Group. The Common Object Request Broker: Architecture and Specification // Technical Report, OMG, July, 1995. – Rev 2.0.
28. Jamnik M., Kerber M., Benzmüller C. Towards New Methods in Proof Planning // Proceedings of the 8th Symposium on the Integration of Symbolic Computation and Mechanized Reasoning. – Scotland, St.Andrews, 2000. – P. 144 – 159.