

ДЕДУКТИВНІ, ІНДУКТИВНІ ТА АНАЛІТИЧНІ МЕТОДИ ПОДАННЯ І ОБРОБКИ КОМП'ЮТЕРНИХ ЗНАТЬ В ІНТЕЛЕКТУАЛЬНИХ СИСТЕМАХ (1. ДЕДУКТИВНІ МЕТОДИ І ЗАСОБИ)

Вступ

В даній роботі розглядаються деякі проблеми штучного інтелекту, а саме комплексний підхід до рішення проблем подання та обробки комп'ютерних знань на базі проведення на ЕОМ логіко-алгебраїчних побудов у формалізованих теоріях. Слід зауважити, що поняття "штучний інтелект" не можна зводити лише до створення пристроїв, які повністю або частково імітують діяльність людини.

Розглянемо і проаналізуємо в загальних рисах деякі проблеми, які доводиться постійно вирішувати людському розумові при рішенні задач, пов'язаних з комп'ютерною обробкою формалізованих знань, що базуються на таких парадигмах: чисельна, аналітична (символьна), дедуктивна, індуктивна та інтеграційна.

Чисельна парадигма відображає засоби і методи наближеного або точного рішення задач чистої або прикладної математики, що базується на побудові скінченної послідовності дій над скінченними множинами чисел.

Аналітична 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 і виникла в середині 1960-х рр. як альтернатива обчислювальній парадигмі у зв'язку з появою ЕОМ такої швидкодії, інформаційної ємності і гнучкості, що стало можливим програмування складних інтелектуальних процесів.

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 додаткові знання беруться з наявних шляхом 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, більша частина яких існує у вигляді систем доведення теорем, оскільки саме логіко-математичний підхід виявляється найбільш релевантним та ефективним при рішенні задач на проведення логічних побудов і використовується в багатьох прикладних галузях.

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

Інтеграційна парадигма "об'єднує" усі вищеназвані парадигми. Її можна поділити на два типи: інтегрування на етапі проектування, тобто ще під час розробки системи передбачається наявність у ній систем різного роду, можливості її ієрархічного нарощування та підключення до неї вже наявних комп'ютерних служб, та інтегрування на етапі експлуатації, тобто комбінування в одну систему вже готових комп'ютерних служб (особливе зацікавлення до розробки таких служб визвало широке застосування мережі Internet для передачі необхідних даних, що, у свою чергу, привело до створення відповідних стандартів).

Що стосується інтеграції на етапі експлуатації, то тут відмітимо, що в рамках проектів Open Mechanized Reasoning Systems [1] та OpenMath [2] розроблено мови специфікацій та спілкування відповідно в галузі доведення теорем та символічних обчислень. І ці мови допускають поєднання.

Найбільш відомим представником підходу до інтеграції на етапі проектування є система Mizar [3]. Іншим представником може бути проект Theorema [4].

В Україні у кінці 1960-х років академіком В.М. Глушковым був запропонований так званий Алгоритм Очевидності (АО) [5]¹, який ще в той час можна було віднести до інтеграційної парадигми, що ставить питання про одночасне проведення досліджень у напрямку розробки формалізованих мов для запису математичних текстів у найсприятливішій для користувача формі, у напрямку еволюційного розвитку поняття машинного (дедуктивного та індуктивного) кроку доведення, створення інформаційного середовища ЕА, яке впливає на очевидність машинного кроку доведення, і в напрямку побудови засобів інтерактивної обробки математичних текстів. Опис поточного стану робіт щодо обробки комп'ютерних знань у відповідності з сучасним баченням Алгоритму Очевидності приводиться у даній роботі. Причому опис стосується тільки дедуктивних методів та засобів. Інші методи та засоби обробки комп'ютерних знань в інтелектуальних системах будуть описані пізніше у ряді статей.

1. Особливості обробки математичних текстів у стилі Алгоритму Очевидності

Відповідно до програми АО, ядром системи автоматизованого пошуку доведення мусить бути процедура встановлення очевидності довідного (верифікованого) твердження в термінах деякого дедуктивного механізму. Таку процедуру необхідно посилити додатковими засобами: мати можливості пошуку допоміжних тверджень чи якої-небудь іншої релевантної інформації, застосовувати алгебраїчні перетворення (або, у сучасній термінології, системи комп'ютерної алгебри), використовувати звичні для людини прийоми доведення та ін. Весь цей інструментарій мусить мати можливість обмінюватися даними за допомогою деякої формалізованої мови.

До мови були сформульовані такі вимоги. Синтаксис і семантика цієї мови повинні бути формалізовані. Вона мусить допускати формулювання аксіом теорії, теорем та їх доведень для забезпечення цілісності текстів і формулювання визначень. Щоб тезаурус мови можна було поповнювати, його повиненно відокремити від граматики мови. Крім того, мова повинна бути близькою до мови математичних публікацій, щоб забезпечити користувачу зручність при створенні й обробці тексту в інтерактивному режимі. Більш того, мова повинна дозволяти формулювати задачі, що не зв'язані безпосередньо з пошуком виведення.

Перший опис такої мови даний у [10]. Остаточний варіант мови, названої TL (Theory Language) [11], з'явився на початку 1970-х рр. [12].

Дедуктивні засоби розвивалися одночасно з мовою у відповідності з такими вимогами: синтаксична форма початкової задачі повинна бути збережена; дедуктивні побудови повинні проводитися в сигнатурі початкової теорії; пошук доведення повинен бути цілеорієнтованим; обробка рівностей (розв'язання рівнянь) повинна бути відокремлена від дедуктивних процесів. У результаті було розроблене числення α -секвенцій [13].

Таким чином, програма АО була зорієнтована на побудову різних складових, необхідних для обробки математичних текстів, записаних формалізованою мовою (тобто процедур для пошуку виведення, символічних обчислень, побудови контрприкладів, застосування визначень та ін.). Ідеї, що лежать в основі програми АО, розроблялися в цілій серії робіт і були частково реалізовані в рамках системи автоматизованої обробки математичних текстів, названої САД.

¹ В. Глушковым у другій половині 60-х років також був ініційований аналогічний підхід до створення родини комп'ютерних алгебр у вигляді ієрархії систем, що розвиваються, виконуючих аналітичні перетворення, серед яких треба відмітити спеціалізовані ЕОМ серії МІР з мовою Аналітик [6], системи Аналітик-79 [7] для СМ ЕОМ 1410 та Аналітик-93 [8] і Аналітик-2000 [9] для ПК типу IBM PC.

На теперішній час розроблена нова, удосконалена, версія мови TL, названа ForTheL (FORmal THEory Language) [14]. Основна мета розробки мови ForTheL – забезпечити початкове середовище для подальшого розвитку ядра Алгоритму Очевидності та підсилити його можливості, що залежать від задачі, яка розв'язується. Відповідно до існуючого підходу до пошуку доведення в стилі Алгоритму Очевидності, оброблюваний математичний текст має бути записаний мовою ForTheL та перетворений потім у так званий ForTheL1-текст, що являє собою цілісний текст. ForTheL1-текст складається з пропозицій, що, з одного боку, є аналогами формул логіки першого порядку, а, з іншого боку, зберігають сигнатуру початкового ForTheL-тексту, його синтаксис і структуру, тобто поділ на такі розділи, як визначення, допоміжні твердження, теорема, яку треба довести. Після перетворення ForTheL-тексту в ForTheL1-текст можна здійснити побудову інформаційного ForTheL1-середовища пошуку доведення.

Для пошуку логічного виведення в ForTheL1-середовищі в [15] був запропонований спеціальний формалізм секвенційного типу.

У результаті використання в цьому формалізмі оригінального поняття а-секвенції (замість звичайної секвенції) вдалося досягти таких переваг:

- розвинути спеціальну техніку роботи з кванторами (яка не вимагає попередньої сколемізації), що допускає пошук виведення в сигнатурі початкової теорії і дозволяє досягти ефективності пошуку, порівняної з ефективністю методів, які вимагають попередньої сколемізації;

- звести довідне твердження до сукупності нових допоміжних тверджень, накопичуючи «рівняння», які треба розв'язати, і не вказуючи при цьому, які терми варто підставляти замість змінних;

- відкладати пошук розв'язку «рівнянь» до довільного моменту часу з метою відокремити його (пошук розв'язку) від дедуктивного процесу;

- вибирати процедуру розв'язку рівнянь (наприклад, застосовувати звичайну уніфікацію, АСа-уніфікацію, Е-уніфікацію), додавати спеціальні правила для рівностей (наприклад, парамодуляцію), застосовувати різні засоби, наявні в системах комп'ютерної алгебри.

Таким чином, важливою рисою секвенційного формалізму є те, що він дає можливість інтегрувати дедукцію і символічні обчислення на рівні логічних числень. Крім того, даний формалізм дозволяє поєднати різні дедуктивні системи: числення Кангера, системи пошуку спрощування, базовані на принципі резолюцій Робінсона, а також зворотний метод Маслоу в його різних інтерпретаціях. Він дає можливість порівнювати зазначені дедуктивні системи [16] і переносити результати, отримані для однієї системи, на іншу.

1.1. Мовні засоби

Мовні засоби відіграють у всьому проекті АО в певному розумінні двоїсту роль. По-перше, нам потрібна придатна мова спілкування, що дозволяє виражати математичні факти і міркування в звичному для математиків виді, а не подавати їх, як це часто необхідно, із самого початку (розв'язку задачі) у рамках деякого точного логічного формалізму – відповідне перетворення мусить виконуватися «автоматично». По-друге, і це, можливо, ще важливіше, роль мови полягає в тому, щоб забезпечити відповідне середовище для розв'язуваної задачі (довідної теореми), яке надалі повинне передаватися процедурі доведення. Середовище може містити необхідні визначення, допоміжні твердження (леми), корисні схеми доведення та ін., а також алгоритми, що дозволяють використовувати всі ці засоби. Середовище, з одного боку, повинне задовольняти деяким умовам замкненості – усе, що використовується, мусить бути описане, а, з іншого боку, вона не повинна бути занадто велика за обсягом. І, нарешті, формування середовища має відбуватися динамічно – залежно від поведінки процедури доведення і поточного стану простору пошуку.

Наші досвід і спостереження показують, що методи, «базовані на мові», виявляються досить ефективними для досягнення згаданої вище мети. Ми сподіваємося, що мова ForTheL дозволить зробити ще один крок у цьому напрямку.

Власне кажучи, ForTheL-текст можна розглядати як сукупність фраз, наділену деякою структурою, що забезпечує певні «правила навігації» у тексті. Тому видається доцільним розрізняти в мові дві в певному розумінні ортогональні складові. Перша — це мова ForTheL-фраз, друга ж визначає композицію всього тексту: групування фраз, іменування розділів, область дії змінних і т.п.

1.2. Синтаксис і семантика ForTheL-тексту

1.2.1. Синтаксис

З погляду синтаксису ForTheL-текст являє собою сукупність розділів. Кожен розділ, у свою чергу, може містити розділи більш низького рівня і/чи фрази. Границі розділів відзначаються роздільниками, що визначає відношення часткового порядку \sim в певному розумінні відношення підпорядкованості \sim на множині розділів даного тексту. Синтаксис роздільників може бути обраний у достатньому ступені довільно, але ми бажаємо діяти згідно з звичайною практикою нумерації, що прийнята в математичних текстах (наприклад, стиль LNCS для LaTeX).

Деякі розділи: \sim теореми, доведення, визначення \sim — відіграють у мові особливу роль. Вони відрізняються спеціальним синтаксисом роздільників, що також відповідає звичайній математичній практиці.

Розділ, що містить тільки фрази, будемо називати простим. Кожна фраза або стверджує, або допускає дещо. Відмінність між твердженнями і припущеннями виражається за допомогою простих синтаксичних засобів.

1.2.2. Семантика

Семантика ForTheL-тексту визначається тією задачею, яку формулює користувач. Візьмемо деякі приклади таких задач: «перевірити коректність даного тексту», «перевірити коректність речення з даного тексту», «знайти контрприклад до твердження з даного тексту», «знайти в тексті інформацію, релевантну ...» і т.п.

Розглянемо, наприклад, другу з перелічених задач. Назвемо речення даного тексту коректним, якщо воно може бути формально виведене зі своїх «попередників». Такий опис вимагає придатного визначення того, що таке «логічний попередник», який можна задати за допомогою перегляду тексту — при цьому роздільники відіграють роль «дорожніх знаків» — і шляхом заміни деяких розділів їх «головними твердженнями». Таким чином, цілий розділ «теорема» можна замінити формулюванням виведення теореми.

1.3. Синтаксис і семантика ForTheL-фрази

Мова ForTheL-фраз містить як власну підмножину мову першого порядку. Крім того, він має істотні особливості: використовується так зване словесне написання, тобто конструкції мови записуються у вигляді слів (природної мови); у мові є так звані поняття, що є аналогами іменників.

Словесне написання дозволяє будувати ForTheL-фрази подібно до фраз природної мови — використовуючи підмети, присудки, доповнення і визначення. ForTheL також забезпечує скорочену символіку, прийняту в математичній логіці.

Логічні зв'язки, а також деякі поняття і предикати є неозначуваними (базовими) синтаксичними одиницями мови ForTheL. Інші синтаксичні одиниці повинні вводитися за допомогою визначень.

Предикатні і функціональні символи в ForTheL'i визначаються у словесній формі; можливі і відповідні символічні позначення. Предикати виражаються за допомогою дієслів і дієприкметників, функції – за допомогою іменників. Таким чином, твердження $P(A) > n$ може бути записане у вигляді power of A is greater than n. Використання словесної форми в рамках символічної не допускається. Більш того, кажучи про словесну форму, будемо далі використовувати термін «шаблон» замість термінів «функціональний символ» чи «предикатний символ».

Корисною особливістю шаблонів предикатів є те, що вони дозволяють розміщати більш одного терма на першому аргументному місці. Така конструкція перетвориться в кон'юнкцію відповідних літералів. $a, b \text{ divide } n$ таким чином перетвориться в $(n | a \wedge n | b)$. Надалі мова буде забезпечувати коректну обробку виразів вигляду $a, b \text{ are equal}$ для симетричних транзитивних бінарних предикатів.

Мова ForTheL регулює вживання артиклів, множини іменників, дієприкметників теперішнього часу. Передбачається поступове поповнення мови загальноживаними виразами.

1.3.1. Поняття в мові ForTheL

Словесне написання обумовлене тим, що ForTheL наближений до природної мови і може розглядатися як «косметичний» засіб. Однак на ньому базується інша ключова особливість мови – поняття. Використання понять, по-перше, відповідає стилю викладу людини, а, по-друге, може допомогти в пошуку виведення при обробці ForTheL-тексту.

Поняття можна розглядати як аналог терма мови першого порядку, але його значенням є не окремий об'єкт, а (параметризована) сукупність об'єктів. Прикладом поняття є relation on X, де змінна X описана як множина (інше поняття мови ForTheL). Сукупність (чи клас) об'єктів не повинна ототожнюватися з множиною, що містить ці об'єкти. З погляду семантики мови ForTheL, множина є окремим об'єктом, так само, як і його елементи, тоді як клас, специфікований поняттям, не є елементом тієї ж предметної області.

Кожному шаблону поняття з n позиціями відповідає характеристичний $(n + 1)$ -місний предикат. Відповідне відношення виражає той факт, що деякий об'єкт належить класу, специфікованому цим поняттям. Предикатний символ може бути описаний безпосередньо в тексті визначення нового шаблону поняття як символ, що його позначає. Для шаблону поняття function from π_1 to π_2 таким предикатним символом може слугувати $\pi : \pi_1 \rightarrow \pi_2$ (за допомогою π ми позначаємо аргументні місця).

1.3.2. Звуження і розширення понять

Можна зробити поняття більш визначеним (тобто звузити відповідний клас) за допомогою спеціальних модифікаторів – атрибутів і пост-умов. Одні атрибути додають обмеження на деякий елемент класу, наприклад, empty set чи number dividing N. Інші додають обмеження на значення деякої унарної функції, визначеної на даному класі, наприклад, set of power K, set of finite power. Неважко помітити, що атрибути є похідними шаблонів предикатів. Граматика мови ForTheL допускає спільне використання декількох атрибутів. Пост-умова – це звичайне речення мови ForTheL, що приєднується до шаблону поняття за допомогою зв'язки such that.

Можливе також розширення поняття шляхом збільшення числа аргументів. Таким чином, якщо визначений шаблон поняття *relation on π* , то неповне поняття *relation* буде задавати клас всіх відношень на всіх множинах даної предметної області, а не на деякій окремій множині.

1.3.3. Змінні та сорти

У ForTheL'i нова змінна описується або за допомогою припущення, у якому змінна прирівнюється до деякого терма, або за допомогою припущення про її належність до класу, що задається деяким поняттям. Обидва твердження можуть бути виражені як явно за допомогою предикатів *is equal to* і *is a* (причому тільки в словесній формі), так і неявно шляхом іменування функцій та понять. Наприклад, твердження *set_A contains x* неявно описує A як змінну, значенням якої є множина. Символ підкреслення позначає іменування.

Термін «сорт» вживається тут у трохи відмінному від загальноприйнятого розуміння та позначає шаблон поняття. Визначаючи нове поняття через інші поняття, ми задаємо ієрархію сортів. Сорт шаблону чи предиката шаблону поняття є послідовність сортів його аргументних місць; сорт шаблону функції є послідовність сортів його аргументних місць разом із сортом значення функції.

У правильно побудованому ForTheL-тексті кожній описаній змінній відповідає її сорт: шаблон поняття, що специфікує клас, якому належить змінна. Таким чином, з'являється можливість додаткового контролю коректності.

У деяких випадках ієрархія сортів може вказати вірний напрямок пошуку виведення. Частковий порядок на множині сортів може бути використаний програмою пошуку доведень для вибору, в першу чергу, тих засновків, що містять поняття, пов'язані з метою.

1.3.4. Терми

Терми в мові ForTheL або об'єкти мови, що можуть знаходитися на аргументних місцях шаблонів предикатів, функцій та понять, можуть мати один з виглядів:

- терм мови першого порядку;
- іменоване поняття, в якого аргументні місця зайняті термами. Іменоване поняття визначає неявно нову змінну, і терм замінюється цією змінною;
- поняття, аргументні місця якого зайняті термами, випереджене лексемою *class of*; значенням терма буде множина, що складається з елементів класу, специфікованого даним поняттям;
- поняття, аргументні місця якого зайняті термами, випереджене лексемою *some* чи *every*; такий терм замінюється новою змінною відповідного сорту і наділяється логічним контекстом, вираженим квантором існування або загальності, що обмежує цю змінну; контекст надалі обробляється в залежності від способу вживання терма. Наприклад, твердження *all divisors of some number are simple* буде перетворене в $\exists n(\text{number}(n) \wedge \forall d(\text{divisor}(d, n) \supset \text{simple}(d)))$.

1.3.5. Використання кванторів

Мова ForTheL дозволяє задавати квантори, що обмежують поняття: *for every simple number_P true that P is odd*. Дане твердження перетвориться як формула з квантором загальності по змінній P, сортом якої є клас простих чисел. Поняття може бути розіменоване: *there exists an even simple number*. У такому випадку з'являється нова змінна.

Для унарних понять у мові ForTheL можуть бути введені синоніми. Твердження *there exists a divisor of N* можна записати так: *N has a divisor*.

1.3.6. Визначення нових шаблонів

У мові ForTheL шаблони понять *class* і *set*, а також шаблони предикатів *is equal to*, *belongs to* і *is a* є неозначуваними. Як зазначено вище, всі інші шаблони і символи визначаються безпосередньо в ForTheL-тексті. Однак при формалізації конкретної теорії можна зручніше розширити перелік неозначуваних шаблонів, ніж зводити їх до теоретико-множинного рівня. Зауважимо, що шаблони характеристичних предикатів понять дозволяють виключити предикат *is a* під час трансляції.

Визначення нових шаблонів — це ForTheL-фрази спеціального вигляду. Зауважимо, що шаблон предиката визначається шляхом вказівки умови істинності предиката, шаблон функції — шляхом прирівнювання її до деякого терма чи вказівкою критерію рівності (можливо, з доведенням функціональності), шаблон поняття — або шляхом звуження деякого поняття, або шляхом вказівки характеристичного відношення. Мова ForTheL допускає також і рекурсивні визначення.

На закінчення зауважимо, що подробиці синтаксису понять і особливості використання мови ForTheL наведені в додатках А і Б.

2. Дедуктивні методи

Однією з особливостей дедукції в стилі АО є те, що пошук доведення виконується в середовищі цілісного ForTheL1-тексту, що є результатом перетворення початкового ForTheL-тексту. Оскільки ForTheL1-текст є структурованим, тобто містить визначення, допоміжні твердження і теорему, яку треба довести, секвенційний підхід, запропонований у [7] для пошуку виведення формул першого порядку, отримує в даній роботі подальший розвиток. Тут пропонується числення, що містить спеціальні правила виведення для роботи з визначеннями і допоміжними твердженнями. Як і раніше, дедуктивний процес у стилі АО має такі складові: зведення (редукція) довідної «мети» до сукупності допоміжних «підцілей»; ефективна обробка кванторів без попередньої сколемізації з метою пошуку доведення в сигнатурі початкової теорії; розв'язок «рівнянь», породжених на стадії редукції мети.

Зауважимо, що перша робота [9] про пошук доведення в стилі АО з'явилася в 1966 р. У ній описувалася евристична процедура пошуку доведення в теорії груп, що допускала використання на формальному рівні методів пошуку доведень, які зустрічаються в математичних публікаціях. Згодом ця формальна техніка була поширена на деякий фрагмент теорії множин. Своє завершення вона отримала у вигляді числення ПДЦ (Пошуку Допоміжних Цілей) [10], що призначалося для перевірки істинності формул класичної логіки першого порядку. Подальший розвиток ПДЦ викликав появу першого представника сімейства секвенційних числень у стилі АО [4]. Щоб визначити місце цих числень серед інших, ми даємо нижче короткий опис наявної в даний час теоретичної бази побудови процедур пошуку доведень теорем.

2.1. Основні підходи до пошуку виведення в класичній логіці першого порядку

Кінець 1950-х і початок 1960-х рр. ознаменовані появою електронних обчислювальних машин такої швидкодії, обсягу пам'яті й оснащеності, що стала можливою автоматизація складних інтелектуальних процесів. Відгуком

на появу обчислювальної техніки такого сорту стала публікація серії робіт, що стосується реалізації числень генценівського типу [19], а також методів пошуку виведення, базованих на результатах Сколема [20] і Ербрана [21]. Подробиці можна знайти в роботах [22 – 25].

Варто зауважити, що в цих перших роботах була дана відповідь на принципове питання про можливість використання обчислювальних машин для побудови математичних міркувань, але недолік ефективних машинно-орієнтованих засобів оптимізації перебору при побудові міркувань перешкоджав отриманню доведень навіть досить простих тверджень класичної логіки першого порядку. Дослідження з поліпшення ефективності запропонованих методів пошуку доведень привели до появи числення Кангера [26] (генценівський підхід) і методу резолюцій Робінсона [27] (сколемівсько-ербранівський підхід). На сьогоднішній день другий підхід дав найбільш ефективні машинно-орієнтовані процедури пошуку виведення для класичної логіки першого порядку, включаючи метод резолюцій Робінсона. (Заслугує на увагу зворотний метод Маслова [28], оскільки він може бути сформульований у термінах резолюції для класичної логіки першого порядку.) Таким чином, дослідження з автоматизації доведення теорем були зосереджені переважно на вдосконаленні можливостей сколемівсько-ербранівського підходу. Зокрема, були запропоновані алгоритми уніфікації, що враховують особливості тієї чи іншої теорії першого порядку з рівністю (A-уніфікація, AC-уніфікація та ін.).

Досліджувалися способи додавання спеціальних правил обернення з рівністю (наприклад, правила парамодуляції) до резолюційних методів. Постійні зусилля в напрямку розвитку сколемівсько-ербранівського підходу привели до появи методу семантичних таблиць, методу графів зв'язків, цілеорієнтованих методів пошуку та ін.

Більш низька ефективність генценівського підходу (у порівнянні зі сколемівсько-ербранівським підходом) може пояснюватися тим, що навіть числення кангерівського типу (і числення ПДЦ [18], зокрема), що відмовляються від обов'язкового виконання попередньої сколемізації, приводять до надлишкового перебору, викликаному можливістю різних порядків застосування логічних (в основному, кванторних) правил і необхідністю подвоєння головних формул при застосуванні деяких з цих правил. У той же час машинно-орієнтовані методи генценівсько-кангерівського типу є відображенням способів доведень, «природніших» для людини.

Крім того, вони дозволяють створювати досить гнучкий інструментарій для підтримки діалогу між користувачем і ЕОМ при інтерактивному пошуку виведення, а також сприяють розумінню процесу машинного доведення людиною. Таким чином, підвищення ефективності пошуку виведення в численні ПДЦ поряд зі збереженням коректності (і, якщо можливо, повноти) дедуктивної обробки «природних» математичних текстів і застосуванням математичних фактів, відомих АО, стає важливою задачею пошуку доведення в рамках АО.

У результаті проведених досліджень було розроблене числення а-секвенцій [13]. Це коректне і повне числення першого порядку, в якому використовується оригінальне поняття припустимої підстановки [29], введене з метою оптимізації додаткових зусиль, зв'язаних з можливістю різних порядків застосування кванторних правил (без попередньої сколемізації). (Пізніше було показано [29], що поняття припустимої підстановки може бути досить просто «вбудовано» у стандартні генценівські числення.)

Помітимо, що поряд з численнями генценівсько-кангерівського типу в рамках АО досліджувалися також числення сколемівсько-ербранівського типу [30].

2.2. Пошук виведення в середовищі математичних ForTheL1-текстів

У даному розділі подане розширення а-секвенційного формалізму з [15] на ForTheL1-тексти, що забезпечує застосування визначень і допоміжних тверджень у ForTheL1-середовищі. Щоб зробити виклад достатньо

незалежним від роботи [15], тут вводяться всі необхідні визначення. Формалізм a -секвенцій використовувався в [15] як для формальної побудови числення gS , так і для неформального опису числення mS . Останнє можна розглядати як числення gS , збагачене правилами застосування визначень і допоміжних тверджень. Для числення, визначеного нижче, збережене позначення « mS ». Також зауважимо, що числення mS при його алгоритмічному проробленні з метою дає цілий спектр різних реалізацій процедур пошуку виведення в дедуктивних системах [34 – 36].

Нагадаємо, що відповідно до сучасного бачення АО обробка цілісного математичного тексту поділяється на три етапи:

- запис початкового математичного тексту у вигляді ForTheL-тексту;
- переклад ForTheL-тексту в ForTheL1-текст;
- пошук доведення в численні mS у середовищі ForTheL1-тексту.

У додатку В наведений приклад з області нестандартного аналізу, що демонструє повний цикл обробки математичного тексту в стилі АО.

2.2.1. Попередні зведення

Основним об'єктом у численні mS є a -секвенція. A -секвенцію можна розглядати як спеціальне узагальнення звичайного поняття секвенції. Ми розглядаємо секвенції, сукцедент яких складається лише з одного об'єкта (названого метою).

У даній роботі ми вважаємо, що класичне числення предикатів першого порядку подане у вигляді секвенційного числення G , описаного в [31].

Ми використовуємо поняття підстановки, визначене в [27]. Підстановочна компонента – це вираження виду t/x , де x – змінна, а t – терм підстановки.

Нехай L – літера, позначимо $\sim L$ її заперечення. Використовуємо вираз $L(t_1, \dots, t_n)$ для позначення того, що t_1, \dots, t_n є список усіх термів (можливо, з повтореннями), що займають аргументні місця в літері L , в тому порядку, в якому вони зустрічаються в ній. Позначимо $F|_y^x$ результат заміни у формулі F змінної x на змінну y .

Крім звичайних змінних, будемо використовувати дві злічених множини спеціальних змінних: невідомі і фіксовані змінні («фіктивні» і «параметри» у термінології [26]).

Впорядкована трійка $\langle w, F, E \rangle$ називається ансамблем, якщо w є послідовність (слово, що складається з невідомих і фіксованих змінних); F – формула першого порядку; E – множина пар термів t_1, t_2 (рівностей виду $t_1 = t_2$).

Назвемо a -секвенцією вираз вигляду $[P], [D], [B], \dots, \langle w_n, P_n, E_n \rangle \Rightarrow \langle w, F, E \rangle$, де $\langle w_1, P_1, E_1 \rangle, \dots, \langle w_n, P_n, E_n \rangle, \langle w, F, E \rangle$ – ансамблі; $[B]$ – список літер, можливо, порожній, $[D]$ і $[P]$ – списки ForTheL-речень, що відповідають визначенням і допоміжним твердженням.

Ансамблі в антецеденті a -секвенції називаються засновками, а ансамбль у сукцеденті – метою даної a -секвенції. Сукупність засновків розглядається як множина; таким чином, порядок засновків неістотний.

Нехай W – множина послідовностей невідомих і фіксованих змінних, s – підстановка. Покладемо $A(W, s) = \{ \langle z, t, w \rangle : z \text{ – змінна } s; t \text{ – терм } s; w \in W, \text{ а } z \text{ розташована в } w \text{ ліворуч від деякої} \}$

фіксованої змінної з t . Тоді s називається допустимою для W , якщо всі змінні s є невідомими змінними, у $A(W, s)$ не існує (різних) елементів

$\langle z_1, t_1, w_1 \rangle, \dots, \langle z_n, t_n, w_n \rangle$ таких, що $t_2 / z_1 \in s, \dots, t_n / z_{n-1} \in s, t_1 / z_n \in s (n > 0) \dots$

Розщеплення деякого ForTheL-речення F за головною логічною зв'язкою і можлива взаємодія його з засновком P_i приводить до породження нових а-секвенцій. Множини E_1, \dots, E_n, E визначають терми, що повинні бути підставлені замість невідомих змінних, щоб перетворити кожен рівність $t_1 = t_2$ з E_1, \dots, E_n, E у тотожність $t = t$ в результаті застосування до E_1, \dots, E_n, E підстановки, обраної визначеним способом. Множини w_1, \dots, w_n, w служать для перевірки допустимості підстановки, отриманої при пошуку доведення. Помітимо, що в будь-якій а-секвенції деякі (чи усі) послідовності w_1, \dots, w_n, w і деякі (чи усі) множини E_1, \dots, E_n, E можуть бути порожніми.

Початкова а-секвенція будується так. Нехай деякий цілісний ForTheL1-текст Txt містить сукупність $[D]$ визначень і сукупність $[P]$ допоміжних тверджень, а T – теорема, яку можна подати в термінах числення G як секвенцію вигляду $P_1, \dots, P_n \Rightarrow F$. Тоді будемо вважати а-секвенцію $[P], [D], [], \langle, P_1, \rangle, \dots, \langle, P_n, \rangle \Rightarrow \langle, F, \rangle$ початковою (відносно T і Txt).

При пошуку виведення в численні mS будується дерево виведення. На початку пошуку воно складається з початкової а-секвенції. В процесі пошуку вузли дерева виведення породжуються відповідно до правил, наведених нижче. Дерево виведення нарощується «зверху вниз». У формулюваннях правил, наведених нижче, MS позначає множину засновків, $[D]$ ($[D_1], [D_2]$) є список визначень (можливо, порожній), $[P]$ ($[P_1], [P_2]$) є список допоміжних тверджень (можливо, порожній). Іноді ми будемо опускати списки $[P], [D]$ у тому випадку, коли вони не використовуються.

Введемо поняття позитивного (негативного) входження літери L у формулу F (позначимо відповідно $F[L^+]$ чи $F[L^-]$) з точністю до рівностей:

– нехай літера $F(\sim F)$ може бути отримана зі $L(t_1, \dots, t_n)$ шляхом заміни термів t_1, \dots, t_n деякими термами t'_1, \dots, t'_n . Будемо говорити, що L має позитивне (негативне) входження в літеру F з точністю до рівностей $t_1 = t'_1, \dots, t_n = t'_n$;

– якщо $F[L^+]$ ($F[L^-]$) з точністю до рівностей $t_1 = t'_1, \dots, t_n = t'_n$ і F_1 є формула, то L має позитивне (негативне) входження (з точністю до рівностей $t_1 = t'_1, \dots, t_n = t'_n$) у формули $F \wedge F_1; F_1 \wedge F; F \vee F_1; F_1 \vee F; F_1 \supset F; \forall xF; \exists xF$;

– якщо $F[L^+]$ ($F[L^-]$) з точністю до рівностей $t_1 = t'_1, \dots, t_n = t'_n$ і F_1 є формула, то L має негативне (позитивне) входження (з точністю до рівностей $t_1 = t'_1, \dots, t_n = t'_n$) у формули $F \supset F_1$ і $\neg F$;

– інших випадків позитивних (негативних) входжень L у F немає.

2.2.2. Правила розщеплення мети (PM)

Правила PM використовуються для елімінації головної логічної зв'язки з ForTheL1-реченням, що знаходиться в меті оброблюваної а-секвенції. Застосування кожного з правил веде до породження нової а-секвенції з єдиною метою (і, можливо, новими засновками). Елімінація ForTheL1-аналогів пропозиційних зв'язок виконується так само, як і в класичній логіці першого порядку, і може бути виражена в термінах похідних правил стандартного числення генценівського типу [31], при цьому $w_1, \dots, w_n, w, E_1, \dots, E_n$ не змінюються. Істотна відмінність від традиційних засобів пошуку виведення генценівського типу спостерігається при обробці кванторів. Ця відмінність відбиває спеціальні способи обробки кванторів, що вивчалися в [13], при яких змінні елімінованих кванторів заміщаються (залежно від вигляду і позиції елімінованого квантора) невідомими чи фіксованими змінними. При цьому w (але не $w_1, \dots, w_n, E_1, \dots, E_n, E$) змінюється і породжуються нові засновки.

Нижче ми використовуємо F' для позначення результату однократного пронесення знаку « \neg » у формулі F відповідно до тотожностей: $\neg(F_1 \vee F_2) \equiv \neg F_1 \wedge \neg F_2$; $\neg(F_1 \wedge F_2) \equiv \neg F_1 \vee \neg F_2$; $\neg(F_1 \supset F_2) \equiv F_1 \wedge \neg F_2$; $\neg\neg F_1 \equiv F_1$; $\neg\forall x F_1(x) \equiv \exists x \neg F_1(x)$; $\neg\exists x F_1(x) \equiv \forall x \neg F_1(x)$.

Пропозиційні правила:

$(\Rightarrow \supset_1)$ -правило

$$\frac{[B], M \Rightarrow \langle w, F \supset F_1, E \rangle}{[B], M, \langle w, F, E \rangle \Rightarrow \langle w, F_1, E \rangle};$$

$(\Rightarrow \supset_2)$ -правило

$$\frac{[B], M \Rightarrow \langle w, F \supset F_1, E \rangle}{[B], M, \langle w, \neg F_1, E \rangle \Rightarrow \langle w, \neg F, E \rangle};$$

$(\Rightarrow \wedge)$ -правило

$$\frac{[B], M \Rightarrow \langle w, F \wedge F_1, E \rangle}{[B], M \Rightarrow \langle w, F, E \rangle \quad [B], M \Rightarrow \langle w, F_1, E \rangle};$$

$(\Rightarrow \vee_1)$ -правило

$$\frac{[B], M \Rightarrow \langle w, F \vee F_1, E \rangle}{[B], M, \langle w, \neg F, E \rangle \Rightarrow \langle w, F_1, E \rangle};$$

$(\Rightarrow \vee_2)$ -правило

$$\frac{[B], M \Rightarrow \langle w, F \vee F_1, E \rangle}{[B], M, \langle w, \neg F_1, E \rangle \Rightarrow \langle w, F, E \rangle};$$

$(\Rightarrow \neg)$ -правило

$$\frac{[B], M \Rightarrow \langle w, \neg F, E \rangle}{[B], M \Rightarrow \langle w, F', E \rangle};$$

де F' є результат однократного пронесення « \neg » у F .

Кванторні правила:

$(\Rightarrow \forall)$ -правило

$$\frac{[B], M \Rightarrow \langle w, \forall x F, E \rangle}{[B], M \Rightarrow \langle w, \bar{x}, F|_{\bar{x}}^x, E \rangle};$$

$(\Rightarrow \exists)$ -правило

$$\frac{[B], M \Rightarrow \langle w, \exists x F, E \rangle}{[B], M, \langle w, \forall x \neg F, E \rangle \Rightarrow \langle w x', F|_{x'}^x, E \rangle};$$

де \bar{x} є нова фіксована змінна.

де x' є нова невідома змінна.

2.2.3. Правила допоміжної мети (DM)

Правила ДМ відбивають той факт, що числення mS орієнтоване на пошук виведення шляхом перетворення речень цілісних математичних ForTheL1-текстів. Правила ДМ можна інтерпретувати в термінах числень генценівського типу як елімінація головних логічних зв'язок з ForTheL1-реченнями, що входять до засновків. (Ці речення породжуються детермінованим чином, починаючи з такого засновку $\langle w_i, P_i, E_i \rangle$, що P_i містить позитивне входження (з точністю до рівностей) ForTheL1-пропозиції F з метою α -секвенції, що є вхідною для правила ДМ. У випадку елімінації головних логічних зв'язувань у засновках залишаються в силі зауваження, зроблені для правил РМ, за винятком зауважень, що стосуються $w_1, \dots, w_n, w, E_1, \dots, E_n, E$.) У результаті застосування правил ДМ утвориться mS ($m > 0$) α -секвенцій з новими цілями $\langle w'_1, F_1, E'_1 \rangle, \dots, \langle w'_m, F_m, E'_m \rangle$ і, можливо, деякими новими (щодо α -секвенції, що є вхідною для правила ДМ) засновками.

Пропозиційні правила:

($\supset_1 \Rightarrow$)-правило

$$\frac{[B], \langle w, F \lfloor L^- \rfloor \supset F_1, E' \rangle, M \Rightarrow \langle w', L, E \rangle}{[B], \langle w, (\neg F) \lfloor L^+ \rfloor E' \rangle, M \Rightarrow \langle w', L, E \rangle \quad [B, \sim L], M \Rightarrow \langle w, \neg F_1, E' \rangle};$$

($\supset_2 \Rightarrow$)-правило

$$\frac{[B], \langle w, F \supset F_1 \lfloor L^+ \rfloor E' \rangle, M \Rightarrow \langle w', L, E \rangle}{[B], \langle w, F_1 \lfloor L^+ \rfloor E' \rangle, M \Rightarrow \langle w', L, E \rangle \quad [B, \sim L], M \Rightarrow \langle w, F, E' \rangle};$$

($\vee_1 \Rightarrow$)-правило

$$\frac{[B], \langle w, F \vee F_1 \lfloor L^+ \rfloor E' \rangle, M \Rightarrow \langle w', L, E \rangle}{[B], \langle w, F_1 \lfloor L^+ \rfloor E' \rangle, M \Rightarrow \langle w', L, E \rangle \quad [B, \sim L], M \Rightarrow \langle w, \neg F, E' \rangle};$$

($\vee_2 \Rightarrow$)-правило

$$\frac{[B], \langle w, F \lfloor L^+ \rfloor \vee F_1, E' \rangle, M \Rightarrow \langle w', L, E \rangle}{[B], \langle w, F \lfloor L^+ \rfloor E' \rangle, M \Rightarrow \langle w', L, E \rangle \quad [B, \sim L], M \Rightarrow \langle w, \neg F_1, E' \rangle};$$

($\wedge_1 \Rightarrow$)-правило

$$\frac{[B], \langle w, F \lfloor L^+ \rfloor \wedge F_1, E' \rangle, M \Rightarrow \langle w', L, E \rangle}{[B], \langle w, F \lfloor L^+ \rfloor E' \rangle, \langle w, F_1, E' \rangle, M \Rightarrow \langle w', L, E \rangle};$$

($\wedge_2 \Rightarrow$)-правило

$$\frac{[B], \langle w, F \wedge F_1 \lfloor L^+ \rfloor E' \rangle, M \Rightarrow \langle w', L, E \rangle}{[B], \langle w, F, E' \rangle, \langle w, F_1 \lfloor L^+ \rfloor E' \rangle, M \Rightarrow \langle w', L, E \rangle};$$

($\neg \Rightarrow$)-правило

$$\frac{[B], \langle w, \neg(F \lfloor L^- \rfloor), E' \rangle, M \Rightarrow \langle w', L, E \rangle}{[B], \langle w, F \lfloor L^+ \rfloor E' \rangle, M \Rightarrow \langle w', L, E \rangle},$$

де F' – результат однократного пронесення « \neg » у F .

Термінальні правила:

$(\Rightarrow \#_1)$ -правило

$$\frac{[B], \langle w, L(t_1, \dots, t_n), E' \rangle, M \Rightarrow \langle w', L(t'_1, \dots, t'_n), E \rangle}{M \Rightarrow \langle w, \#, E'' \rangle},$$

де $E'' = E' \cup E \cup \{t_1 = t'_1, \dots, t_n = t'_n\}$; $L(t_1, \dots, t_n)$, $L(t'_1, \dots, t'_n)$ – літери.

$(\Rightarrow \#_2)$ -правило

$$\frac{[B_1, L(t_1, \dots, t_n), B_2], M \Rightarrow \langle w', L(t'_1, \dots, t'_n), E \rangle}{[B_1, L(t_1, \dots, t_n), B_2], M \Rightarrow \langle w', \#, E' \rangle},$$

де $E' = E \cup \{t_1 = t'_1, \dots, t_n = t'_n\}$; $L(t_1, \dots, t_n)$, $L(t'_1, \dots, t'_n)$ – літери.

Кванторні правила:

$(\forall \Rightarrow)$ -правило

$$\frac{[B], \langle w, \forall x(F \lfloor L^+ \rfloor), E' \rangle, M \Rightarrow \langle w' L, E \rangle}{[B], \langle w x', F \lfloor_x^+ \rfloor, E' \rangle, \langle w, \forall x F, E' \rangle, M \Rightarrow \langle w', L, E \rangle},$$

де x' є нова невідома змінна.

$(\exists \Rightarrow)$ -правило

$$\frac{[B], \langle w, \exists x(F \lfloor L^+ \rfloor), E' \rangle, M \Rightarrow \langle w', L, E \rangle}{[B], \langle w \bar{x}, F \lfloor_{\bar{x}}^+ \rfloor, E' \rangle, M \Rightarrow \langle w', L, E \rangle},$$

де \bar{x} є нова фіксована змінна.

2.2.4. Правило застосування визначення і правило застосування допоміжного твердження

Структурування ForTheL1-текстів, тобто виділення в них так званих змістовних розділів (визначень, тверджень та ін.), дає можливість включити в числення mS правило застосування визначення (ПВ) і правило застосування допоміжного твердження (ДТ). Ці правила можна розглядати як окремі випадки правила ОЦ. Вони є аналогами природних прийомів роботи з визначеннями і допоміжними твердженнями при доведенні теорем.

Правило ПВ формулюється з урахуванням структури розділу «визначення» ForTheL-тексту. Відповідно до синтаксису мови ForTheL розділ «визначення» складається з двох частин: дескрипції і дефініції. Дескрипція є сукупність припущень, що задовольняє умові замкненості: для кожної змінної, що зустрічається в деякому припущенні дескрипції, у тій же дескрипції знайдеться припущення, що описує цю змінну. Дефініція є ForTheL-речення вигляду $A(x_1, \dots, x_n) \text{ IFF } F(x_1, \dots, x_n)$, де x_1, \dots, x_n – змінні, причому x_i ($i = 1, \dots, n$) має входження в дескрипцію. Формула $A(x_1, \dots, x_n)$, як правило, є атомом, а $F(x_1, \dots, x_n)$ є формула, що не містить атома А (ми обмежуємося випадком нерекурсивних визначень).

Розділ «визначення» з дескрипцією, що складається з припущень F_1, \dots, F_k , і дефініцією виду

$$A(x_1, \dots, x_n) \quad \text{IFF} \quad F(x_1, \dots, x_n) \quad \text{можна} \quad \text{розглядати} \quad \text{як} \quad \text{формулу}$$

$\forall x_1 \dots x_n (F_1 \wedge \dots \wedge F_k \supset (A(x_1, \dots, x_n) \equiv F(x_1, \dots, x_n)))$. Оскільки застосування визначення в математичній практиці виконується звичайно так, ніби використовувалася одна з формул $\forall x_1 \dots x_n (F_1 \wedge \dots \wedge F_k \supset (A(x_1, \dots, x_n) \supset F(x_1, \dots, x_n)))$ або вигляду $\forall x_1 \dots x_n (F_1 \wedge \dots \wedge F_k \supset (F(x_1, \dots, x_n) \supset A(x_1, \dots, x_n)))$, ми вважаємо, що розділ «визначення» складається з формул зазначеного вище вигляду.

Допоміжне твердження являє собою розділ ForTheL-тексту, що складається з припущень і висновку. Висновок можна розглядати як формулу вигляду $F(x_1, \dots, x_n)$, де x_1, \dots, x_n – змінні, що входять до припущення твердження. Сукупність припущень задовольняє зазначеній вище умові замкненості або може бути перетворена таким чином, щоб ця умова виконувалася (тут ми не розглядаємо відповідні перетворення, докладніше про це, наприклад, у [30], [32]).

У наведеному нижче формулюванні правила ПВ $F[L^+]$ позначає $A(x_1, \dots, x_n) \equiv F(x_1, \dots, x_n)$.

ПВ-правило

$$\frac{[P], [D_1, \forall x_1 \dots x_n (F_1 \wedge \dots \wedge F_k \supset F[L^+]), D_2], [B], M \Rightarrow \langle w', L, E \rangle}{A_1 \Rightarrow \langle w', L, E \rangle \quad A_1' \Rightarrow \langle x_1' \dots x_n', F_1 \wedge \dots \wedge F_k, \rangle}$$

Тут A_1 позначає

$$[P], [D_1, \forall x_1 \dots x_n (F_1 \wedge \dots \wedge F_k \supset (A(x_1, \dots, x_n) \supset F(x_1, \dots, x_n))), D_2], [B], \langle x_1' \dots x_n', A(x_1, \dots, x_n) \supset F(x_1, \dots, x_n), \rangle, M.$$

$$A_1' \text{ позначає } [P], [D_1, \forall x_1 \dots x_n (F_1 \wedge \dots \wedge F_k \supset (A(x_1, \dots, x_n) \supset F(x_1, \dots, x_n))), D_2], [B, \sim L], M,$$

$$[P], [D_1, \forall x_1 \dots x_n (F_1 \wedge \dots \wedge F_k \supset (A(x_1, \dots, x_n) \supset F(x_1, \dots, x_n))), D_2],$$

якщо L має позитивне входження в $F(x_1, \dots, x_n)$;

$$A_1 \text{ позначає } [P], [D_1, \forall x_1 \dots x_n (F_1 \wedge \dots \wedge F_k \supset (F(x_1, \dots, x_n) \supset A(x_1, \dots, x_n))), D_2],$$

$$[B], \langle x_1' \dots x_n', F(x_1, \dots, x_n) \supset A(x_1, \dots, x_n), \rangle, M,$$

а A_1' позначає $[P], [D_1, \forall x_1 \dots x_n (F_1 \wedge \dots \wedge F_k \supset (F(x_1, \dots, x_n) \supset A(x_1, \dots, x_n))), D_2], [B, \sim L], M,$

якщо L має позитивне входження в $A(x_1, \dots, x_n)$.

ДТ-правило

$$\frac{[P_1, \forall x_1 \dots x_n (F_1 \wedge \dots \wedge F_k \supset F[L^+]), P_2], [D], [B], M \Rightarrow \langle w', L, E \rangle}{A_2 \Rightarrow \langle w', L, E \rangle \quad A_2' \Rightarrow \langle x_1' \dots x_n', F_1 \wedge \dots \wedge F_k, \rangle}$$

Тут A_2 позначає $[P_1, \forall x_1 \dots x_n (F_1 \wedge \dots \wedge F_k \supset F[L^+]), P_2], [D], [B], \langle x_1' \dots x_n', F[L^+] \rangle, M,$

а A_2' позначає $[P_1, \forall x_1 \dots x_n (F_1 \wedge \dots \wedge F_k \supset F[L^+]), P_2], [D], [B, \sim L], M.$

2.2.5. Правило додавання засновку (ДЗ)

Дане правило впливає на все дерево пошуку цілком. Після кожного застосування ($\forall \Rightarrow$)-правила ($\exists \Rightarrow$)-правила) до антецедентів всіх а-секвенцій поточного дерева, що містять засновки з формулою, яка включає виділені входження $\langle w, \forall x(F[L^+]), E' \rangle$ ($\langle w, \exists x(F[L^+]), E' \rangle$), додається новий засновок $\langle wx', F \uparrow_{x'}^x [L^+], E' \rangle$ ($\langle w\bar{x}, F \uparrow_{\bar{x}}^x [L^+], E' \rangle$).

2.2.6. Аксиоми

Аксиомами є а-секвенції вигляду $[P], [D], [B], M \Rightarrow \langle w, \#, E \rangle$, де $\#$ позначає порожню формулу.

2.2.7. Дерево виведення

Твердження Т, що потрібно довести, подається як змістовний розділ «теорема» ForTheL1-тексту. У цьому розділі умови (припущення) і висновок теореми відокремлені один від одного. Початкова (відносно Т) а-секвенція будується таким чином, що припущення попадають в антецедент, а висновок – у сукцедент. (Інша частина Txt ForTheL1-тексту являє собою множину визначень і допоміжних тверджень.)

Застосовуючи правила виведення «зверху вниз» до початкової а-секвенції, а потім до її «нащадків» та ін., ми отримуємо у підсумку дерево виведення (щодо теореми Т, яку потрібно довести, і ForTheL1-тексту Txt, що складає її «оточення»). Дерево виведення Tr називається деревом доведення для вхідної а-секвенції тоді і тільки тоді, коли

- а) кожний лист дерева Tr є аксіомою;
- б) існує уніфікатор s усіх рівностей з Tr;
- в) s допустиме (у розумінні даної роботи) для множини всіх послідовностей фіксованих і невідомих змінних, що знаходяться в листах дерева Tr.

На будь-якому етапі пошуку виведення може бути виконана перевірка можливості перетворення поточного дерева виведення в дерево доведення. Якщо побудова дерева доведення здійснюється в інтерактивному режимі, то таку перевірку може ініціювати користувач. Для виконання перевірки можуть бути використані засоби побудови найбільш загального уніфікатора.

2.3. Коректність і повнота числення mS

Вище зазначалося, що кожне ForTheL1-речення можна розглядати як аналог деякої формули класичної логіки першого порядку. Це дозволяє будувати формули, що відповідають таким одиницям ForTheL1-тексту, як довідна теорема, визначення, допоміжне твердження, а також працювати з цілісним ForTheL1-текстом як з множиною формул першого порядку. Таким чином, такі вирази, як «несуперечність ForTheL1-тексту», «дана теорема є логічним наслідком даного ForTheL1-тексту», «загальнозначність» (довідної теореми) розуміються однозначно і не вимагають спеціального визначення семантики мови ForTheL1. З огляду на сказане, сформулюємо основні результати про числення mS.

Твердження 1. ForTheL1-теорема Т є логічний наслідок несуперечливого ForTheL1-тексту Txt (не утримуючого Т) тоді і тільки тоді, коли в mS може бути побудоване дерево доведення з початковою секвенцією відносно Т і Txt.

Твердження 2. ForTheL1-теорема T загальнозначна тоді і тільки тоді, коли в численні mS може бути побудоване дерево виведення з початковою секвенцією відносно T.

Відзначимо також, що досить багатий набір правил числення mS дозволяє будувати різноманітні стратегії пошуку доведення, що моделюють звичайні математичні доведення, а при пошуку доведення в інтерактивному режимі користувач має можливість впливати на його процес. Якщо стратегія (незалежно від того, передбачає вона участь чи людини ні) забезпечує вичерпний пошук, то твердження 1 і 2 гарантують її коректність і повноту.

3. Приклади

Приклад А. Синтаксис понять

Тут подано фрагмент граматики мови ForTheL, що описує синтаксис понять.

```

term:      [ ALL | SOME | CLASS OF ] concept
|         function                               // power of S
|         symbolic_term                         // P(S)
concept:   left_attr free_concept right_attr
[ SUCH THAT statement ]
free_concept: C_TOK [ _ var ]
|           C_TOK [ _ var ] C_TOK term
|           C_TOK [ _ var ] C_TOK term C_TOK term
left_attr:  left_constr
right_attr: right_constr
|          OF F_TOK term                       // of power N
|          OF left_constr F_TOK                // of nonzero power
|          OF F_TOK right_constr              // of power less than N
left_constr: [ NOT ] P_TOK                     // empty ...
right_constr: [ NOT ] EQUAL TO term
|           [ NOT ] BELONGING TO term
|           [ NOT ] P_TOK term                 // ... dividing N
|           [ NOT ] P_TOK P_TOK term           // ... less than N

```

Приклад Б. Приклад ForTheL-тексту

У правильно побудованому ForTheL-тексті, наведеному нижче, введені деякі визначення з теорії множин.

Definition 1. Element of A is a set belonging to set $_A$.

Definition 2. A is empty IFF wrong that exists an element of set $_A$.

There exists unique empty set.

Definition 3. Empty_set (0) is equal to any empty set.

Definition 4. Let A, B be sets.

A is a subset of B ($A \text{ \textit{sub} B}$) IFF all elements of A belong to B.

Lemma 1. Each set has a subset.

Proof. 0 is a subset of all sets. QED.

Definition 5.

Powerset of A (PS[A]) is equal to the class of subsets of set A .

Definition 6 "Tarski". Let A be a set.

A is finite IFF for every not empty subset K of PS[A] there exists subset S of A belonging to K such that S is not a subset of any element of K not equal to S

Приклад В. Дедукція в стилі АО

Тут наведений приклад обробки математичного тексту відповідно до схеми АО. Подане доведення (але не пошук доведення) теореми з [33]: «Кожна замкнена підмножина компактної множини компактна». Якщо діяти відповідно до схеми, необхідно «занурити» дану теорему в деякий математичний текст. Тут наводиться частина тексту, що містить ті припущення і розділи, які використовуються у виведенні, поданому нижче. Таким чином, визначення деяких понять (наприклад, нескінченно близької точки, функції ST) у текст не включені: ці визначення не використовувалися як засновки ПВ-правила. Для простоти викладу ми використовуємо подання секвенцій у стилі Генцена, а не у вигляді а-секвенцій, «переносячи» обробку рівностей і послідовностей змінних у коментарі.

Теорема, яку треба довести, розглядається як частина наступного цілісного математичного ForTheL-тексту.

Section 1. "Sets and Relations"

Lemma 1:1. Every subset of any set is a set.

Lemma 1:2. Let U be a subset of set V .

Then all elements of U are the elements of V .

Lemma 1:3. For any set Y ST[Y] is a set.

Lemma 1:4. For any subset U of set V

ST[U] is a subset of ST[V].

Lemma 1:5. Every subset of every subset of any set U is a subset of U .

Section 2. "Topological Spaces"

Lemma 2:1. Any topological_space is a set.

Definition 2:1.

Let Y be a set. Any point of Y is an element of Y . Definition 2:2. Let M be a subset of topological_space Y . M is

M is

closed IFF for any point Q of Y if there exists point of ST[M] infinitesimally_close to Q then Q is a point of

M .

Definition 2:3. Let M be a subset of topological_space Y .

M is compact IFF every point of ST[M] is infinitesimally_close to some point of M .

Theorem 2:1. Let K be a compact subset of topological_space X . Then any closed subset of K is compact.

В результаті синтаксичних перетворень даного ForTheL-тексту отримуємо наступний ForTheL1-текст.

Section 1.

Lemma 1:1.

Let V be a set. Let W be a subset of V . W is a set.

Lemma 1:2. Let V be a set. Let U be a subset of V .

If t is an element of U then t is an element of V .

Lemma 1:3. Let Y be a set. ST[Y] is a set.

Lemma 1:4. Let V be a set. If U is a subset of V then ST[U] is a subset of ST[V].

Lemma 1:5. For all U, V, W if U is a subset of V and V is a subset of W then U is a subset of W.

Section 2.

Lemma 2:1. If Y is a topological_space then Y is a set.

Definition 2:1. Let Y be a set.

If P is a point of Y then P is an element of Y.

If P is an element of Y then P is a point of Y.

Definition 2:2.

Let Y be a topological_space. Let M be a subset of Y.

If M is closed then for any Q (if Q is a point of Y then (if there exists P such that (P is a point of ST[M] and P is infinitesimally_close to Q) then Q is a point of M)).

If for any Q (if Q is a point of Y then (if there exists P such that (P is a point of ST[M] and P is infinitesimally_close to Q) then Q is a point of M)) then M is closed.

Definition 2:3.

Let Y be a topological_space. Let M be a subset of Y.

If M is compact then for any P (if P is a point of ST[M] then there exists Q such that (Q is a point of M and P is infinitesimally_close to Q)).

If for any P (if P is a point of ST[M] then there exists Q such that (Q is a point of M and P is infinitesimally_close to Q)) then M is compact.

Theorem 2:1.

Let X be a topological_space. Let K be a subset of X.

Let K be compact. Let F be a subset of K. Let F be closed.

F is compact.

Розглянемо доведення. Теоремі, що доводиться, відповідає наступна початкова секвенція.

$[D], [P], [], < \bar{X} \text{ is a } T_S \wedge \bar{K} \text{ is a subset of } \bar{X} \wedge \bar{F} \text{ is a subset of } \bar{K} \wedge \bar{K} \text{ is compact} \wedge \bar{F} \text{ is closed}, > \Rightarrow < \bar{F} \text{ is compact}, >$.

Далі ми використовуємо скорочення $T_S \approx$ відповідно для «topological_space» і «infinitesimally_close». Списки $[D]$ і $[P]$ являють собою середовище доведення, перший включає всі визначення, а другий – всі леми з наведеного вище ForTheL1-тексту. Надалі для стислості викладу ми будемо опускати списки $[D]$ і $[P]$, вказуючи однак ті визначення і допоміжні твердження, до яких застосовуються правила ПВ і ДТ. Помітимо, що коли застосовується правило ПВ (ДТ), створюється копія відповідного визначення (леми), при цьому виконується перейменування змінних. Нові змінні, що зустрічаються далі в тексті, мають верхні індекси, що позначають номер відповідного визначення (леми).

Позначимо Π сукупність засновків « $\bar{X} \text{ is a } T_S, \bar{K} \text{ is a subset of } \bar{X}, \bar{F} \text{ is a subset of } \bar{K}, \bar{K} \text{ is compact}, \bar{F} \text{ is closed}$ » з антецедента початкової секвенції (надкреслені змінні – це фіксовані змінні). До визначення 2:3 застосовне правило ПВ. У результаті породжуються дві секвенції:

1. $[\sim \bar{F} \text{ is compact}], \Pi \Rightarrow Z_1^2 \text{ is a } T_S \wedge Z_2^2 \text{ is a subset of } \bar{X}$.
2. $\Pi, \forall Z_3^2 (Z_3^2 \text{ is a point of } ST(Z_2^2) \supset \exists Z_4^2 (Z_4^2 \text{ is a point of } Z_2^2 \wedge Z_3^2 \approx Z_4^2)) \supset Z_2^2 \text{ is compact} \Rightarrow \bar{F} \text{ is compact}$.

З огляду на двоїстий характер ForTheL1-речень, будемо використовувати стандартні логічні зв'язки для скорочення запису. По тій же причині ми не дублюємо формули при застосуванні кванторних правил. Отже, застосовуючи правило РМ до секвенції 1, отримуємо нові секвенції:

$$1.1. [\sim \bar{F} \text{ is compact}], \Pi \Rightarrow Z_1^2 \text{ is a T_S.}$$

$$1.2. [\sim \bar{F} \text{ is compact}], \Pi \Rightarrow Z_2^2 \text{ is a subset of } \bar{X}.$$

Секвенція 1.1 довідна, відповідна підстановка $\{\bar{X} / Z_1^2\}$. Застосовуючи правило ДМ до 2, маємо

$$2.1. \Pi, Z_2^2 \text{ is compact} \Rightarrow \bar{F} \text{ is compact.}$$

$$2.2. [\sim \bar{F} \text{ is compact}], \Pi \Rightarrow \forall Z_3^2 (Z_3^2 \text{ is a point of } ST(Z_2^2) \supset \exists Z_4^2 (Z_4^2 \text{ is a point of } Z_2^2 \wedge Z_3^2 \approx Z_4^2)).$$

Секвенція 2.1 довідна, відповідна підстановка $\{\bar{F} / Z_2^2\}$. Застосовуючи далі правила РМ до 2.2, отримуємо

$$2.2.1. [\sim \bar{F} \text{ is compact}], \Pi, \bar{Z}_3^2 \text{ is a point of } ST(\bar{F}), \forall Z_4^2 \neg (Z_4^2 \text{ is a point of } \bar{F} \wedge \bar{Z}_3^2 \approx Z_4^2) \Rightarrow Z_4^2 \text{ is a point of } \bar{F}.$$

$$2.2.2. [\sim \bar{F} \text{ is compact}], \Pi, \bar{Z}_3^2 \text{ is a point of } ST(\bar{F}), \forall Z_4^2 \neg (Z_4^2 \text{ is a point of } \bar{F} \wedge \bar{Z}_3^2 \approx Z_4^2) \Rightarrow \bar{Z}_3^2 \approx Z_4^2.$$

Розглянемо секвенцію 2.2.2. Позначимо сукупність її засновків Π_1 . Правило ПВ застосовне до визначення 2:3. Застосовуючи до 2.2.2 правила числення \neg (спочатку ПВ, а потім ДМ), отримуємо секвенції:

$$2.2.2.1. [\sim \bar{F} \text{ is compact}, \sim \bar{Z}_3^2 \approx Z_4^2], \Pi_1 \Rightarrow Z_5^2 \text{ is a T_S.}$$

$$2.2.2.2. [\sim \bar{F} \text{ is compact}, \sim \bar{Z}_3^2 \approx Z_4^2], \Pi_1 \Rightarrow Z_6^2 \text{ is a subset of } \bar{X}.$$

$$2.2.2.3. [\sim \bar{F} \text{ is compact}, \sim \bar{Z}_3^2 \approx Z_4^2], \Pi_1 \Rightarrow Z_7^2 \text{ is a point of } ST(\bar{K}).$$

$$2.2.2.4. [\sim \bar{F} \text{ is compact}], \Pi_1, \bar{Z}_8^2 \text{ is a point of } \bar{K}, Z_7^2 \approx \bar{Z}_8^2 \Rightarrow \bar{Z}_3^2 \approx Z_4^2.$$

Секвенції 2.2.2.1, 2.2.2.2, 2.2.2.4 довідні, відповідна (загальна) підстановка є $\{\bar{X} / Z_5^2, \bar{K} / Z_6^2, \bar{Z}_3^2 / Z_7^2, \bar{Z}_8^2 / Z_4^2\}$. Помітимо, що з'явилися два нових засновки: $\bar{Z}_8^2 \text{ is a point of } \bar{K}$ і $\bar{Z}_3^2 \approx \bar{Z}_8^2$. Приймаючи до уваги ці засновки (тобто застосовуючи правило ДЗ) і отримавши підстановку, можемо подати секвенцію 2.2.1 у наступному вигляді: $[\sim \bar{F} \text{ is compact}], \Pi_1, \bar{Z}_8^2 \text{ is a point of } \bar{K}, \bar{Z}_3^2 \approx \bar{Z}_8^2 \Rightarrow \bar{Z}_3^2 \text{ is a point of } \bar{F}.$

Тепер правило ПВ застосуємо до визначення 2:2. Послідовне застосування правил ПВ, ДТ і РМ дає

$$2.2.1.1. [\sim \bar{F} \text{ is compact}, \sim \bar{Z}_8^2 \text{ is a point of } \bar{F}], \Pi_1, \bar{Z}_8^2 \text{ is a point of } \bar{K}, \bar{Z}_3^2 \approx \bar{Z}_8^2 \Rightarrow Z_1^2 \text{ is a T_S.}$$

$$2.2.1.2. [\sim \bar{F} \text{ is compact}, \sim \bar{Z}_8^2 \text{ is a point of } \bar{F}], \Pi_1, \bar{Z}_8^2 \text{ is a point of } \bar{K}, \bar{Z}_3^2 \approx \bar{Z}_8^2 \Rightarrow Z_2^2 \text{ is a subset of } \bar{X}.$$

$$2.2.1.3. [\sim \bar{F} \text{ is compact}, \sim \bar{Z}_8^2 \text{ is a point of } \bar{F}], \Pi_1, \bar{Z}_8^2 \text{ is a point of } \bar{K}, \bar{Z}_3^2 \approx \bar{Z}_8^2 \Rightarrow Z_3^1 \text{ is a point of } \bar{X}.$$

2.2.1.4. [$\sim \bar{F}$ is compact], Π_1, \bar{Z}_8^2 is a point of $\bar{K}, \bar{Z}_3^2 \approx \bar{Z}_8^2, Z_3^1$ is a point of $Z_2^1 \Rightarrow \bar{Z}_8^2$ is a point of \bar{F} .

2.2.1.5. [$\sim \bar{F}$ is compact, $\sim \bar{Z}_8^2$ is a point of \bar{F}], Π_1, \bar{Z}_8^2 is a point of $\bar{K}, \bar{Z}_3^2 \approx \bar{Z}_8^2 \Rightarrow Z_4^1 \approx \bar{Z}_8^2$.

2.2.1.6. [$\sim \bar{F}$ is compact, $\sim \bar{Z}_8^2$ is a point of \bar{F}], Π_1, \bar{Z}_8^2 is a point of $\bar{K}, \bar{Z}_3^2 \approx \bar{Z}_8^2 \Rightarrow Z_4^1$ is a point of $ST(\bar{F})$.

Секвенції 2.2.1.1, 2.2.1.4, 2.2.1.5, 2.2.1.6 довідні, відповідна підстановка $\{\bar{X}/Z_1^1, \bar{Z}_8^2/Z_3^1, \bar{F}/Z_2^1, \bar{Z}_3^2/Z_4^1\}$.

Розглянемо секвенцію 1.2. Вона може бути подана так:

[$\sim \bar{F}$ is compact], $\Pi \Rightarrow \bar{F}$ is a subset of \bar{X} . Застосовуючи до леми 1:5 (транзитивність включення) правило ДТ, а потім правила РМ, маємо

1.2.1. [$\sim \bar{F}$ is compact, $\sim \bar{F}$ is a subset of \bar{X}], $\Pi \Rightarrow Z_1^t$ is a subset of Z_2^t .

1.2.2. [$\sim \bar{F}$ is compact, $\sim \bar{F}$ is a subset of \bar{X}], $\Pi \Rightarrow Z_2^t$ is a subset of Z_3^t .

1.2.3. [$\sim \bar{F}$ is compact], Π, Z_1^t is a subset of $Z_3^t \Rightarrow \bar{F}$ is a subset of \bar{X} .

Секвенція 1.2 довідна ($\{\bar{F}/Z_1^t, \bar{K}/Z_2^t, \bar{X}/Z_3^t\}$). Секвенція 2.2.1.2 може бути доведена аналогічно.

Далі розглянемо 2.2.2.3. Застосовуючи побудовану підстановку, маємо [$\sim \bar{F}$ is compact; $\sim \bar{Z}_3^2 \approx \bar{Z}_8^2$], $\Pi_1 \Rightarrow \bar{Z}_3^2$ is a point of $ST(\bar{K})$. Помітимо, що Π_1 містить засновок \bar{Z}_3^2 is a point of $ST(\bar{F})$. Позначимо антецедент даної секвенції [B_1], Π_1 . До визначення 2:1 застосовне правило ПВ. У результаті маємо

2.2.2.3.1. [$B_1, \sim \bar{Z}_3^2$ is a point of $ST(\bar{K})$], $\Pi_1 \Rightarrow Z_1^3$ is a set.

2.2.2.3.2. [$B_1, \sim \bar{Z}_3^2$ is a point of $ST(\bar{K})$], $\Pi_1 \Rightarrow Z_2^3$ is an element of Z_1^3 .

2.2.2.3.3. [B_1], Π_1, Z_2^3 is a point of $Z_1^3 \Rightarrow \bar{Z}_3^2$ is a point of $ST(\bar{K})$.

Секвенція 2.2.2.3.3 доказова ($\{\bar{Z}_3^2/Z_2^3, ST(\bar{K})/Z_1^3\}$). Тепер 2.2.2.3.1 може бути подана як

[$B_1, \sim \bar{Z}_3^2$ is a point of $ST(\bar{K})$], $\Pi_1 \Rightarrow ST(\bar{K})$ is a set.

Секвенція 2.2.2.3.2 тепер має вигляд [$B_1, \sim \bar{Z}_3^2$ is a point of $ST(\bar{K})$], $\Pi_1 \Rightarrow \bar{Z}_3^2$ is an element of $ST(\bar{K})$.

Розглянемо першу із зазначених секвенцій. Позначимо її антецедент [B_2], Π_2 . Правило ДТ застосовне до леми 1:3.

2.2.2.3.1.1. [$B_2, \sim ST(\bar{K})$ is a set], $\Pi_2 \Rightarrow Z_1^{13}$ is a set.

2.2.2.3.1.2. [B_2], $\Pi_2, ST(Z_1^{13})$ is a set $\Rightarrow ST(\bar{K})$ is a set.

Секвенція 2.2.2.3.1.2 доказова ($\{\bar{K}/Z_1^{13}\}$). Правило ВУ застосовне до 2.2.2.3.1.1 і леми 1:1. Маємо

2.2.2.3.1.1.1. [$B_2, \sim ST(\bar{K})$ is a set, $\sim \bar{K}$ is a set], $\Pi_2 \Rightarrow Z_1^{11}$ is a set.

2.2.2.3.1.1.2. [$B_2, \sim ST(\bar{K})$ is a set, $\sim \bar{K}$ is a set], $\Pi_2 \Rightarrow Z_1^{I3}$ is a subset of Z_1^{I1} .

2.2.2.3.1.1.3. [$B_2, \sim ST(\bar{K})$ is a set], Π_2, Z_2^{I1} is a set $\Rightarrow \bar{K}$ is a set.

Секвенція 2.2.2.3.1.1.2 довідна ($\{\bar{K} / Z_2^{I1}, \bar{X} / Z_1^{I1}\}$).

Застосовуючи підстановку до 2.2.2.3.1.1.1, маємо

[$B_2, \sim ST(\bar{K})$ is a set, $\sim \bar{K}$ is a set], $\Pi_2 \Rightarrow \bar{X}$ is a set.

У результаті застосування правила ДТ до леми 2:1 отримуємо

2.2.2.3.1.1.1.1. [$B_2, \sim ST(\bar{K})$ is a set, $\sim \bar{K}$ is a set, $\sim \bar{X}$ is a set], $\Pi_2 \Rightarrow Z_1^0$ is a T_S.

2.2.2.3.1.1.1.2. [$B_2, \sim ST(\bar{K})$ is a set, $\sim \bar{K}$ is a set], Π_2, Z_1^0 is a set $\Rightarrow Z_1^{I1}$ is a set.

Наведені вище секвенції довідні ($\{\bar{X} / Z_1^0\}$). Позначимо антецедент 2.2.2.3.2 через [B_3], Π_3 .

Застосовуючи правило ДТ до леми 1:2, маємо

2.2.2.3.2.1. [$B_3, \sim \bar{Z}_3^2$ is an element of $ST(\bar{K})$], $\Pi_3 \Rightarrow Z_1^{I2}$ is a subset of Z_2^{I2} .

2.2.2.3.2.2. [$B_3, \sim \bar{Z}_3^2$ is an element of $ST(\bar{K})$], $\Pi_3 \Rightarrow Z_3^{I2}$ is an element of Z_1^{I2} .

2.2.2.3.2.3. [B_3], Π_3, Z_3^{I2} is an element of $Z_2^{I2} \Rightarrow \bar{Z}_3^2$ is an element of $ST(\bar{K})$.

Секвенція 2.2.2.3.2.3 довідна ($\{ST(\bar{K}) / Z_2^{I2}, \bar{Z}_3^2 / Z_3^{I2}\}$). Застосування ДТ до 2.2.2.3.2.1 і леми 1:4 дає секвенції

2.2.2.3.2.1.1. [$B_3, \sim \bar{Z}_3^2$ is an element of $ST(\bar{K})$, Z_1^{I2} is a subset of $ST(\bar{K})$], $\Pi_3 \Rightarrow Z_1^{I4}$ is a subset of Z_2^{I4} .

2.2.2.3.2.1.2. [$B_3, \sim \bar{Z}_3^2$ is an element of $ST(\bar{K})$], $\Pi_3, ST(Z_1^{I4})$ is a subset of $ST(Z_2^{I4}) \Rightarrow Z_1^{I2}$ is a subset of $ST(\bar{K})$,

які доказові ($\{\bar{F} / Z_1^{I4}, \bar{K} / Z_2^{I4}, ST(\bar{F}) / Z_1^{I2}\}$).

Щоб довести 2.2.2.3.2.2, застосуємо правило ПВ до визначення 2:1.

2.2.2.3.2.2.1. [$B_3, \sim \bar{Z}_3^2$ is an element of $ST(\bar{K})$, $\sim \bar{Z}_3^2$ is an element of $ST(\bar{F})$], $\Pi_3 \Rightarrow Z_3^3$ is a set.

2.2.2.3.2.2.2. [$B_3, \sim \bar{Z}_3^2$ is an element of $ST(\bar{K})$, $\sim \bar{Z}_3^2$ is an element of $ST(\bar{F})$], $\Pi_3 \Rightarrow Z_4^3$ is a point of Z_3^3 .

2.2.2.3.2.2.3. [$B_3, \sim \bar{Z}_3^2$ is an element of $ST(\bar{K})$], Π_3, Z_4^3 is an element of $Z_3^3 \Rightarrow \bar{Z}_3^2$ is an element of $ST(\bar{F})$.

Секвенції 2.2.2.3.2.2.2 і 2.2.2.3.2.2.3 довідні ($\{ST(\bar{F}) / Z_3^3, \bar{Z}_3^2 / Z_4^3\}$).

В результаті застосування ДТ до 2.2.2.3.2.2.1 і леми 1:3 маємо

2.2.2.3.2.2.1.1. [$B_3, \sim \bar{Z}_3^2$ is an element of $ST(\bar{K})$, $\sim \bar{Z}_3^2$ is an element of $ST(\bar{F})$, $\sim ST(\bar{F})$ is a set],

$\Pi_3 \Rightarrow Z_3^{I3}$ is a set.

2.2.2.3.2.2.1.2. [$B_3, \sim \bar{Z}_3^2$ is an element of $ST(\bar{K})$, $\sim \bar{Z}_3^2$ is an element of $ST(\bar{F})$], $\Pi_3, ST(Z_3^{I3})$ is a set $\Rightarrow ST(\bar{F})$ is a set.

Секвенція 2.2.2.3.2.2.1.2 довідна ($\{\bar{F} / Z_2^{l3}\}$).

Щоб довести 2.2.2.3.2.2.1.1, треба застосувати правило ДТ до леми 1:1. Секвенція 2.2.1.3 доводиться шляхом застосування правила ПО (до визначення 2:1), ДТ (до леми 1:2), а потім ПВ (до визначення 2:1).

Зрозуміло, щоб висновок у численні m , наведений вище, став таким, якими бувають доведення в звичайних математичних публікаціях, його варто відредагувати. При цьому доцільно «сховати» кроки виведення, пов'язані, наприклад, з переходом від поняття «точка» до поняття «елемент», а також ті, що мають відношення до обробки теоретико-множинних понять.

Висновки

Описані вище мовні і дедуктивні особливості Алгоритму Очевидності дозволяють сполучати засіб пошуку доведень АО та засіб символічних обчислень інших, зовнішніх систем. Така перевага АО виявляється завдяки особливому способу обробки математичних текстів, що враховує як виразні засоби мови ForTheL, так і особливості пошуку виведення в логічному численні.

Оскільки основні логічні і теоретико-множинні відносини є базовими в мові ForTheL, ця мова придатна для подання будь-якого роду знань (не тільки математичних), формалізованих у рамках класичної логіки. ForTheL-текст може досить легко скласти як людина, так і машинна програма, та передати його потім для обробки програмі доведення теорем чи будь-якій іншій програмі або безпосередньо, або через мережу.

Мова може бути використана як засіб написання і перевірки математичних статей та формальних специфікацій, для виконання перевірки моделей і т.д. Пропонується застосовувати його як універсальний інтерфейс у базах декларативних математичних знань і як інструмент інтеграції комп'ютерних математичних служб.

Що стосується дедуктивних засобів АО, то числення mS , як і його прототип gS , можна розглядати як базу для подальшого розвитку АО в напрямі підвищення ефективності і створення інтерактивного інструментарію для пошуку доведень теорем і автоматизації міркувань.

На даний час розроблені граматики для російської та англійської версій мови ForTheL, а також реалізована програма, що перетворює ForTheL-тексти в ForTheL1-тексти. Реалізоване ядро оригінального прувера, базованого на описаному вище a -секвенційному формалізмі, використовує ForTheL як мову подання вхідних текстів і отриманих доведень.

У найближчому майбутньому передбачається завершити реалізацію першої версії системи автоматизації дедукції, яка проводить такий ланцюжок трансформацій:

```
ForTheL-текст, у якому виділена теорема, яка доводиться=====>
(використовуючи синтаксичний ForTheL-аналізатор)
=====> Замкнутий ForTheL1-текст =====>
(використовуючи прувер з розширеними можливостями)
=====> Машинне ForTheL1-доведення виділеної теореми =====>
(використовуючи редактор ForTheL1-доведення)
=====> Доведення теореми в звичному для людини вигляді
```

Поточний стан робіт у цьому напрямку, а також деякі особливості системи САД можна знайти в роботах [34 – 42].

СПИСОК ЛІТЕРАТУРИ

1. <http://www.uug.dist.unige.it/omrs/>
2. <http://www.openmath.org/design.txt>
3. <http://mizar.uwb.edu.pl/>

4. <http://www.theorema.org/>
5. Глушков В.М. Некоторые проблемы теории автоматов и искусственного интеллекта // Кибернетика. – 1970. – № 2. – С. 3 – 13.
6. Аналитик (алгоритмический язык для описания вычислительных процессов с использованием аналитических преобразований) // Глушков В.М., Боднарчук В.Г., Гринченко Т.А., Дородницына А.А., Клименко В.П., Летичевский А.А., Погребинский С.Б., Стогий А.А., Фишман Ю.С. // Кибернетика. – 1971. – № 3. – С. 102 – 134.
7. Аналитик-79 / Глушков В.М., Гринченко Т.А., Дородницына А.А., Драха А.М., Клименко В.П., Погребинский С.Б., Савчук О.Н., Фишман Ю.Ф., Царюк Н.П.: Препр. / АН УССР. Ин-т кибернетики. – К.: 1983. – № 12. – 73 с.
8. Аналитик-93 / Морозов А.А., Клименко В.П., Фишман Ю.С., Бублик Б.А., Горовой В.Д., Калина Е.А. // Кибернетика и системный анализ. – 1995. – № 5. – С. 127 – 157.
9. Аналитик-2000 / Морозов А.А., Клименко В.П., Фишман Ю.С., Ляхов А.Л., Кондрашов С.В., Швалюк Т.Н. // Математические машины и системы. – 2001. – № 1, 2. – С. 66 – 99.
10. О языке для описания формальных теорий / Глушков В.М., Костырко В.Ф., Летичевский А.Л., Ануфриев Ф.В., Асельдеров З.М. // Теоретическая кибернетика. – 1970. – Вып. 3. – С. 4 – 31.
11. 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.
12. A brief historical sketch on Kiev school of automated theorem proving / Degtyarev A., Kapitonova Yu., Letichevsky A., Lyaletski A., Morokhovets M. // Proc. Of the 2nd International THEOREMA Workshop. – Austria, Linz: RISC. – 1998. – P. 151 – 156.
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. Vershinin K., Paskevich A. ForTheL – the language of formal theories // IJ Information Theories and Applications.
15. Degtyarev A., Lyaletski A., Morokhovets M. Evidence Algorithm and Sequent Logical Inference Search // Lect. Notes Artif. Intell. – 1999. – P. 44 – 61.
16. Лялецкий А.В. Секвенциальный формализм и дедуктивные системы для классической логики первого порядка // Proc. of Intern. Conf. «Logic and Applications». – Новосибирск. – 2000.
17. Об одном алгоритме поиска доказательств теорем в теории групп / Ануфриев Ф.В., Федюрко В.В., Летичевский А.Л., Асельдеров З.М., Дидух И.И. // Кибернетика. – 1966. – № 1. – С. 23 – 29.
18. Ануфриев Ф.В. Алгоритм поиска доказательств теорем в логических исчислениях // Теория автоматов. – К.: ИК АН УССР. – 1969. – С. 3 – 26.
19. Gentzen G. Untersuchungen über das Logische Schliessen // Math. Z. – 1934. – P. 176 – 210.
20. Skolem T. Logisch-kombinatorische Untersuchungen über die Erfüllbarkeit oder Beweisbarkeit mathematischer Satze // Skriftner utgit ar Videnskapsselskaper i Kristiania. – 1920. – P. 4 – 36.
21. Herbrand J. Recherches sur la Théorie de la Démonstration // Travaux de la Société de Sciences et de Lettres de Varsovie, Classe III, Sciences Mathématiques et Physiques. – 1930. – P. 3 – 36.
22. Davis M., Putnam H. A computing procedure for quantification theory // Journ. of the ACM. – 1960. – P. 201 – 215.
23. Gilmore P. A proof method for quantification theory; its justification and realization // I.B.M.J. Res. Devl. – 1960. – P. 28 – 35.
24. Prawitz D. An improved proof procedure // Theoria. – 1960. – P. 102 – 139.
25. Wang H. Towards mechanical mathematics // I.B.M.J. Res. Devl. – 1960. – P. 2 – 22.
26. Kanger S. Simplified proof method for elementary logic // Comp. Program. and Form. Sys.: Stud. in Logic. – Amsterdam: North-Holl., Publ. Co. – 1963. – P. 87 – 93.
27. Robinson J. A machine-oriented logic based on resolution principle // Journ. of the ACM. – 1965. – P. 23 – 41.
28. Маслов С.Ю. Обратный метод установления выводимости в классическом исчислении предикатов // Докл. Академии Наук СССР. – 1964. – С. 3 – 36.
29. Lyaletski A. Gentzen calculi and admissible substitutions // Actes préliminaires du Simposium Franco-Soviétique "Informatika-91". – Grenoble, France, 1991. – P. 99 – 111.
30. Атаян В.В., Мороховец 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.
31. Gallier J. Logic for computer science: foundations of Automatic Theorem Proving // New York: Harper and Row, Inc. – 1986.
32. Атаян В.В., Вершинин К.П. К формализации некоторых приемов поиска вывода // Автоматизация обработки математических текстов. – Киев: ИК АН УССР, 1980. – С. 36 – 52.
33. Дэвис М. Прикладной нестандартный анализ. – М.: Мир, 1986.
34. Lyaletski A. About one specificity of the inverse method // Abs. On the Web-site of the Logic Colloquium 2001, Kurt Godel Society. – Vienna, Austria. – 2001. – August. – P. 1.
35. Lyaletski A. Quantifier Handling Issues in Computer-oriented intuitionistic calculation // Advances in soft computing, Proc. of the Int. conference IIS'2001. – Poland: Physica-Verlag, a Springer-Verlag Co. – 2001. – P. 15.
36. Lyaletski A., Paskevych A. Goal-driven inference search in classical prepositional logic // Proc. of the International Workshop "STRATEGIES-2001". – Siena, Italy. – 2001. – P. 65 – 74.
37. Verchinine K., Degtyarev A., Lyaletski A. and Paskevich A. SAD, a System for Automated Deduction: a Current State // Proc. of Intern. Workshop Automath'2002. – Edinburgh, Great Britain. – 2002. – P. 10.
38. System of Automated Deduction (SAD): Linguistic and Deductive Peculiarities / Lyaletski A., Verchinine K., Degtyarev A. and Paskevich A. // "Advances in Soft Computing", Proc. of the Int. Conference IIS'2002. – Poland: Physica-Verlag, A Springer-Verlag Co. – 2002.
39. Особенности обработки математических текстов в системе автоматизированной дедукции (САД) / Асельдеров З.М., Вершинин К.П., Дегтярев А.И., Лялецкий А.В., Паскевич А.Ю. // Международный научно-теоретический журнал «Искусственный интеллект» (Труды III Международной научно-практической конференции "Искусственный интеллект – 2002"), – Кацивели, Украина, 2002.
40. Особенности обработки математических текстов в системе автоматизированной дедукции (САД) / Асельдеров З.М., Вершинин К.П., Дегтярев А.И., Лялецкий А.В., Паскевич А.Ю. // Сборник тезисов докладов III Международной научно-практической конференции "Искусственный интеллект'2002". — Кацивели, Украина: НИИ МВС ТРТУ. – Таганрог, Россия. – 2002.
41. Алгоритм Очевидности и обработка формализованных математических текстов / Вершинин К.П., Дегтярев А.И., Лялецкий А.В., Мороховец М.К., Паскевич А.Ю. // Международный научно-теоретический журнал «Проблемы управления и информатики». – 2002. – № 6.

42. Linguistic Tools and Deductive Technique of the System for Automated Deduction / Aselderov Z., Verchinin K., Degtyarev A., Lyaletski A., Paskevich A. and Pavlov A. // Proc. of the 3rd International Workshop on the Implementation of Logics. – Tbilisi, Georgia. – 2002.