УДК 004.421

М. М. Повідайчик, Д. О. Майорський, І. Я. Шпонтак (ДВНЗ «Ужгородський нац. ун-т»)

РОЗРОБКА УТИЛІТИ АВТОМАТИЗОВАНОГО ФОРМУВАННЯ МАТЕМАТИЧНИХ ВИРАЗІВ НА ОСНОВІ ЛІНГВІСТИЧНОГО АНАЛІЗУ ТЕКСТУ

The article deals with the problem of automated formation of mathematical expressions on the basis of linguistic analysis of the text. It is described the utility for the MS Word processor, which, using the knowledge base in the Visual Prolog language and using the OMath object, generate some mathematical expressions that are described in natural language.

Розглядається задача автоматизованого формування математичних виразів на основі лінгвістичного аналізу тексту. Описано утиліту для текстового процесора MS Word, яка, використовуючи базу знань на мові Visual Prolog, за допомогою об'єкта OMath генерує деякі математичні вирази, описані природною мовою.

1. Вступ. У зв'язку із стрімким зростанням інформаційних потоків актуалізується проблема їхнього автоматизованого опрацювання, зокрема, розробляються методи лінгвістичного аналізу текстової інформації [1]. Проблемі аналізу текстової інформації присвятили дослідження такі вітчизняні та зарубіжні вчені, як: Д. Поспєлов, С. Осуга, Х. Уено, М. Ісідзука, Н. Хомський, А. Гладкий, О. Литвиненко та ін. Проте задача аналізу математичних текстів, поданих природною мовою, на даний час потребує додаткового дослідження. Ця проблема актуалізується у зв'язку з появою комп'ютерних систем перетворення аудіо у текст, потребами дистанційного навчання та ін.

2. Постановка задачі. Розробити комп'ютерну систему лінгвістичного аналізу фрагменту тексту, поданого у текстовому процесорі MS Word, який містить описання математичних виразів природною мовою, та перетворити його у відповідний об'єкт OMath.

3. Викладення основного матеріалу. Історично склалося так, що математичні символи, операції, функції мають досить різноманітну структуру та можуть передаватися природною мовою із певною невизначеністю. Так, один і той же символ, розміщений у різних частинах формули чи співвідношення може мати різний зміст; для розуміння математичного тексту, що передається природною мовою, необхідно враховувати контекст, інтонацію та ін. Тому задача інтерпретації математичних виразів є важливою складовою досліджень у галузі штучного інтелекту.

Опишемо розроблену утиліту «Текст_Формула», яка дозволяє перетворювати деякий математичний текст у об'єкт OMath. У редакторі MS Word 2010 цей об'єкт має такі методи: BuildUp, ConvertToLiteralText, ConvertToMathText, ConvertToNormalText, Linearize, Remove; а також такі властивості: AlignPoint, Application, ArgIndex, ArgSize, Breaks, Creator, Functions, Justification, NestingLevel, Parent, ParentArg, ParentCol, ParentFunction, ParentOMath, ParentRow, Range, Type.

Використання зазначених властивостей і методів створює зручний інструментарій для автоматизованої побудови математичних виразів. При цьому можна використовувати два основні підходи:

- конструювання формули за допомогою властивостей Functions об'єкта OMath;

- формування «лінійного» вигляду формули та перетворення її у «професіональний» вид за допомогою метода BuildUp.

У розробленій системі використовується другий підхід, при цьому побудова «лінійного» вигляду формули має такі особливості:

- більшість базових символів (числа, латинські символи та ін.) та операцій (додавання, множення та ін.) передаються звичним чином;

- деякі об'єкти (степінь, індекс та ін.) передаються символами ASCII;

- для деяких виразів (квадратний корінь, інтеграл та ін.) використовується кодування Unicode;

- довідкова система VBA та сайт MS Office [2] надають часткову інформацію про об'єкт OMath.

Зважаючи на особливості описання математичних виразів природною мовою, складність та неоднозначність розпізнавання математичного тексту, для побудови «лінійного» вигляду формули використовується програма на мові Visual Prolog [3]. Пролог відноситься до декларативних мов програмування і його особливістю є те, що у пролог-програмі описуються зв'язки між елементами деякої предметної області за допомогою фактів та правил у вигляді предикатів першого порядку, тобто будується певна база знань. Побудована таким чином система дозволяє давати відповіді на певні запити, базуючись на методі резолюцій. Також слід відзначити зручність роботи у системі Пролог зі списками, рекурсивними структурами, символьними виразами.

Розглянемо деякі особливості розробленої програми.

Вхідні дані – текст, який описує математичний вираз природною мовою; вихідні дані – «лінійна форма» формули, придатна до відображення у середовищі MS Word за допомогою об'єкта OMath.

Представлення даних. Математичний вираз представляється у префіксній формі та описується рекурентною структурою

```
вираз = цифра(integer);
    латинська(symbol);
    рівно(вираз, вираз);
    сума(вираз, вираз);
    різниця(вираз, вираз);
    добуток(вираз, вираз);
    частка(вираз, вираз);
    мінус(вираз);
    корінь(вираз);
    степінь(вираз, вираз);
    індекс(вираз, вираз)
```

Цільовий предикат «старт» має такий вигляд:

```
старт :-
    consult("Формула.txt"),
    текст(Текст),
    текст_список(Текст, Слова),
    унарний_мінус(Слова, Слова1),
    список_вираз(Слова1, Вираз),
    вираз_формула(Вираз, Формула),
    retractall(_),
    assert(текст(Формула)),
    save("Формула.txt"), !.
```

У описаному правилі предикат «текст_список» перетворює вхідний текст у список слів, предикат «унарний_мінус» опрацьовує випадки використання знаку «мінус» як унарної операції, предикат «список_вираз» перетворює список слів на структуру «вираз», а предикат «вираз_формула» будує «лінійний» вигляд формули. Розглянемо детальніше головний предикат «список_вираз», який описаний процедурою, що містить 17 правил:

```
список_вираз(Список, рівно(Вираз1, Вираз2)) :-
    розділити("дорівнює", 0, Список, Список1, Список2),
    список_вираз(Список1, Вираз1),
    список_вираз(Список2, Вираз2), !.
список_вираз(Список, сума(Вираз1, Вираз2)) :-
    розділити("плюс", 0, Список, Список1, Список2),
    список_вираз(Список1, Вираз1),
    список_вираз(Список2, Вираз2), !.
список_вираз(Список, різниця(Вираз1, Вираз2)) :-
    розділити("мінус", 0, Список, Список1, Список2),
    список_вираз(Список1, Вираз1),
    список_вираз(Список2, Вираз2), !.
список_вираз(["дріб" | Хвіст], частка(Вираз1, Вираз2)) :-
    розділити("ділити", 0, Хвіст, Список1, Список2),
    список_вираз(Список1, Вираз1),
    список_вираз(Список2, Вираз2), !.
список_вираз(["ун_мінус", Елемент], мінус(Вираз)) :-
    список_вираз([Елемент], Вираз), !.
список_вираз(["ун_мінус", "дріб" | Хвіст], мінус(Вираз)) :-
    список_вираз(["дріб" | Хвіст], Вираз), !.
список_вираз(["ун_мінус", Цифра, Буква | Хвіст],
добуток(Вираз1, Вираз2)) :-
    одноцифрове_число(Цифра, Цифра1),
    Вираз1 = мінус(цифра(Цифра1)),
    латинська_буква(Буква, _),
    список_вираз([Буква | Хвіст], Вираз2), !.
список_вираз(["ун_мінус", Буква1, Буква2 | Хвіст],
добуток(Вираз1, Вираз2)) :-
```

```
    латинська_буква(Буква1, Буква3),
    Вираз1 = мінус(латинська(Буква3)),
    латинська_буква(Буква2, _),
    список_вираз([Буква2 | Хвіст], Вираз2), !.
список_вираз(["ун_мінус" | Хвіст], мінус(Вираз)) :-
    список_вираз(Хвіст, Вираз), !.
список_вираз([Буква1, Буква2 | Хвіст],
добуток(Вираз1, Вираз2)) :-
    латинська_буква(Буква1, Буква3),
    Вираз1 = латинська(Буква3),
    not(одноцифрове_число(Буква2, _)),
    список_вираз([Буква2 | Хвіст], Вираз2), !.
список_вираз([Цифра, Буква | Хвіст],
добуток(Вираз1, Вираз2)) :-
    одноцифрове_число(Цифра, Цифра1),
    Вираз1 = цифра(Цифра1),
    список_вираз([Буква | Хвіст], Вираз2), !.
список_вираз([Буква, "квадрат"], степінь(Вираз1, цифра(2))) :-
    латинська_буква(Буква, Буква1),
    Вираз1 = латинська(Буква1), !.
список_вираз(["корінь", Буква], корінь(Вираз)) :-
    латинська_буква(Буква, Буква1),
    Вираз = латинська(Буква1), !.
список_вираз(["корінь", Цифра], корінь(Вираз)) :-
    одноцифрове_число(Цифра, Цифра1),
    Вираз = цифра(Цифра1), !.
список_вираз([Буква, Цифра], індекс(Вираз1, Вираз2)) :-
    латинська_буква(Буква, Буква1),
    Вираз1 = латинська(Буква1),
    одноцифрове_число(Цифра, Цифра1),
    Вираз2 = цифра(Цифра1), !.
список_вираз([Буква], Вираз) :-
    латинська_буква(Буква, Буква1),
    Вираз = латинська(Буква1), !.
список_вираз([Цифра], Вираз) :-
    одноцифрове_число(Цифра, Цифра1),
    Вираз = цифра(Цифра1), !.
```

За допомогою приведених правил список слів задає структуру «вираз» у вигляді дерева, вершиною якого є операція чи співвідношення з найнижчим пріоритетом. Так, у результаті цільового запиту

```
Текст = "ікс один дорівнює мінус а плюс два бе",
текст_список(Текст, Слова),
унарний_мінус(Слова, Слова1),
список_вираз(Слова1, Вираз).
```

буде сформовано

```
Вираз = рівно(індекс(латинська("x"), цифра(1)),
сума(мінус(латинська("a")),
добуток(цифра(2), латинська("b"))))
```

У процедурі використовується допоміжний предикат «розділити», який розбиває список слів на два підсписки. При цьому враховується вкладеність виразів. Наприклад, у цільовому запиті

```
Список = ["дріб","один","плюс","два","ділити","три","плюс","чотири"],
розділити("плюс", 0, Список, Список1, Список2).
```

аналізується вираз з двома операціями «плюс», але коректне розбиття на доданки можливе лише за другою операцією:

```
Список1=["дріб","один","плюс","два","ділити","три"],
Список2=["чотири"]
```

Предикат «вираз_формула» формує зі структури «вираз» формулу у «лінійному» вигляді. Наприклад, цільовий запит

```
Текст = "a квадрат плюс корінь це",
текст_список(Текст, Слова),
список_вираз(Слова, Вираз),
вираз_формула(Вираз, Формула).
```

поверне:

```
Формула=a^2+ChrW(8730)c
```

Тут значення «ChrW(8730)» повертає у VBA знак квадратного кореня, заданого у системі кодування Unicode.

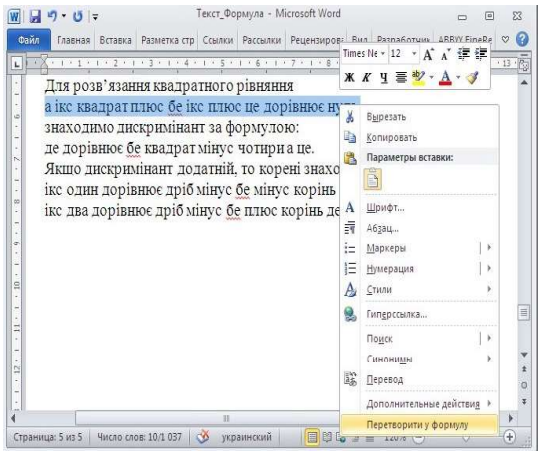

Рис. 1. Текст, що містить математичні вирази, передані природною мовою

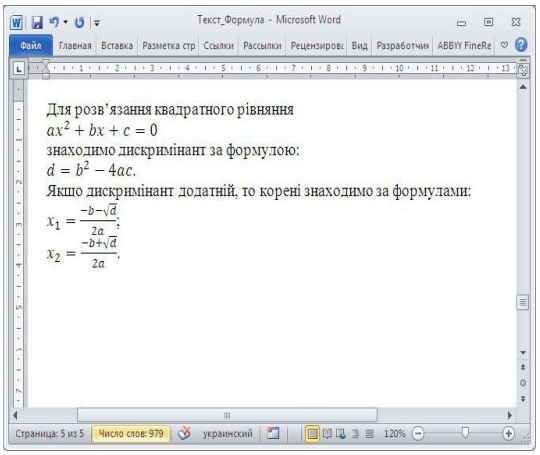

Рис. 2. Текст, опрацьований утилітою «Текст_Формула»

Отже, утиліта «Текст_Формула» дозволяє перетворювати деякі математичні вирази, передані природною мовою, у об'єкти OMath текстового редактора MS Word. На рис. 1-2 наведено результат використання утиліти для тексту, що описує загальний розв'язок квадратного рівняння.

4. Висновки. Розроблена утиліта «Текст_Формула» дозволяє автоматизувати створення деяких математичних виразів у середовищі MS Word за допомогою об'єктів OMath. Описана програма має демонстраційний характер, оскільки розв'язує відносно невеликий клас задач. Але навіть для такого класу виникає багато підзадач, пов'язаних з неоднозначністю передачі математичних виразів природною мовою. Тому в подальшому планується вивчення загальних підходів оптимального представлення даних, ефективних алгоритмів, їх перетворення та ін.

1. *Вавіленкова А.І.* Методи та алгоритми автоматизованого формування логіко-лінгвістичних моделей текстової інформації [Текст]: автореф. дис. ... канд. техн. наук: 05.13.06 / А.І. Вавіленкова; НАН України, Ін-т пробл. мат. машин і систем. – К., 2010. – 20 с.
2. Объект OMath (Word) [Електронний ресурс]. – URL: https://docs.microsoft.com/ru-ru/office/vba/api/word.omath
3. Visual Prolog [Електронний ресурс]. – URL: https://www.visual-prolog.com/

Одержано 26.10.2018