5. Х а р т м а н Ф. Обыкновенные дифференциальные уравнения. М., 1970. 6. Ц и м б а л В.Н. Некоторые неклассические сингулярно возмущенные задачи // Методы малого параметра и их применение: Тез. лекций и кратких научн. сообщ. Всесоюз. школы-семинара. Минск, 1982. С. 118. 7. Ц и м б а л В.М. Змішана задача для сингулярно збуреного псевдопараболічного рівняння // Вісн. Львів. ун-ту. Сер. мех.-мат. Вип. 40. С.3-5. 8. Ц и м б а л В. Нелокальна задача для деяких сингулярно збурених рівнянь у частинних похідних // Нові підходи до розв"язання диференц. рівнянь: Тези доп. Всеукр. наук. конф.-м. Дрогобич, Дрогобич, 1994. С. 180. 9. Ц и м б а л В.Н. Нелокальная задача для сингулярно возмущенных эллиптических и параболических уравнений // Нелинейные задачи математической физики и задачи со свободной границей: Тез. 8-й республ. конф., м.Донецьк. Донецк, 1991. С. 121.

Стаття надійшла до редколегії 24.02.95

УДК 681.3.06

М.О.Дзіковська, О.В.Костів

ЗАСТОСУВАННЯ ОБ"ЄКТНО-ОРІЄНТОВАНОГО ПІДХОДУ ДО ОБРОБКИ ДЕРЕВОВИДНИХ СТРУКТУР ДАНИХ

Дерева належать до найуживаніших абстрактних структур даних, оскільки рекурсивність є природною характеристикою багатьох елементів реального світу. Тому алгоритми опрацювання деревовидних структур викликають зацікавлення як у теоретиків, так і у практиків.

З огляду на широке використання об"єктно-орієнтованого програмування /ООП/, цікавою є задача застосування об"єктно-орієнтованого підходу до роботи з абстрактними структурами даних. Засобами такого стилю програмування можна реалізувати концепцію абстрактних типів даних /АТД/ на базі абстрактних структур даних /АСД/.

Проектування програм за принципом "зверху вниз" передбачає використання різних рівнів деталізації опису алгоритмів. Кожному рівневі відповідає певний набір базових операцій, композиції яких використовуються для зображення алгоритмів на цьому рівні.

Найчастіше розрізняють три рівні опису алгоритмів, які грунтуються на різних формах зображення даних:

1/ рівень абстрактних структур даних; 2/ рівень даних алгоритмічної мови; 3/ рівень реалізації в конкретному комп"ютері.

Для роботи з кореневими впорядкованими навантаженими деревами /далі деревами/ у праці [1] запропонований набір операцій, який містить понад 40 операторів обробки дерев як абстрактних структур даних. Цей набір можна взяти за основу операцій, визначаючи АТД для роботи з деревами.

Прагматичний метод абстрагування [2] передбачає опис значення даних та операцій засобами мов програмування. Тому в пропонованій праці описані результати застосування підходу АТД до деревовидних структур даних у разі їх відображення у структури даних Паскаля та реалізації цього підходу засобами ООП.

Дерева можна відображати у послідовні і зв"язані структури зберігання. У Паскалі послідовні структури зображаються векторами, зв"язані - списками.

Досліджені два векторні та одне списокове зображення. За векторні зображення обраний двоїнковий запис для дерев довільної структури і бездужковий польський префіксний запис для бінарних дерев; за списокове зображення - двозв"язний список для бінарних дерев.

Для кожного зображення дерева створено відповідний об"єкт: brtree для двоїнкового, bintree для бездужкового та dyntree для спискового зображення. Усі три об"єкти мають подібну структуру. Для зображення даних використовуються чотири поля, які характеризують поточний стан дерева: поле зображення дерева tree /типу символьний масив TreeArr для векторних зображень або типу вказівник на вузол ptr для спискового зображення/, поле поточної позиції в дереві pos /типу індексу масиву TreeInd або вказівника/, поле навантаження поточного вузла curnode /символьного типу/, поле результату suc /логічного типу, що набуває значення true, якщо остання операція над об"єктом була успішною і false у протилежному випадку/. Крім цього, для векторних зображень введено ще поле dist з цілочисельним значенням, яке задає відстань між останнім вузлом піддерева, визначеного деяким вузлом, і правим братом цього вузла. Ця відстань є постійною для конкретного векторного зображення. Урахування цього факту через введення додаткового поля дає змогу уніфікувати деякі операції над деревами.

Для створення екземплярів об"єктного типу потрібні конструктори. Для векторних зображень створений один конструктор Init(t), де t - вхідний параметр типу TreeArr. Він заносить значення t у поле зображення дерева tree та ініціює поле dist.

Для списокових зображень розроблені кілька конструкторів: Create_br(t) / t - вхідний параметр типу brtree/ та Create_bin(t) / t - вхідний параметр типу bintree/ створюють списокове зображення з відповідних векторних зображень; Create_search(tr) / tr - вхідний параметр типу treeArr створює дерево пошуку за масивом, сортуючи навантаження вузлів за їхніми кодами;

Create_dyn(td)(id- вхідний параметр типу ptr) створює дерево за відповідним описовим зображенням td.

Усі конструктори ініціюють поля pos та suc, встановлюючи поточним корінь дерева.

До базових операцій над деревами у модулі належить перехід до заданого вузла Gotonode(p) / p - вхідний параметр типу TreeInd для векторних і ptr для списового зображень/. Ця операція змінює поточний вузол, і всі алгоритми повинні змінювати поле pos тільки через неї.

Для векторних зображень базовою також є операція пошуку піддерева Searchtree(b,e) / b -вхідний параметр типу treeInd, e - вихідний параметр того ж типу/; b задає корінь піддерева. Результатом операції є вказівник на останній символ піддерева, який заноситься в e. Операція реалізується по-різному для лінійного і безлінійкового зображень. Для списового зображення вона не виокремлюється, бо піддерево у цьому випадку цілковито визначається вказівником на його вершину.

Операція руху вниз Down(i) здійснює перехід до i-го сина поточного вузла. Вона також належить до базових, хоча у векторних зображеннях використовує процедуру пошуку піддерева.

Операція руху вверх Up здійснює перехід до батька поточного вузла. Цей метод реалізується по-різному для кожного типу зображення.

Усі інші операції реалізуються через чотири вищезгадані (Gotonode(p), Searchtree(b,e), Down(i), Up).

У моделі також реалізовані деякі основні операції над деревами. Зокрема, це операції руху по дереву, оформлені як процедури переходу до наступного Next та попереднього Pred за обходом вузлів, а також операції пересування до лівого Leftshift та правого Rightshift братів поточного вузла.

Операції перетворення типів - це функції, що характеризують поточний вузол і дерево в цілому. До них належать: функція логічного типу Leaf, яка перевіряє, чи є поточний вузол термінальним /значенням є true, якщо вузол термінальний, і false - у протилежному випадку/; функції цілочисельного типу для обчислення кількості синів поточного вузла Decnum, кількості термінальних Termdecnum і нетермінальних Intermdecnum серед них; функції цілочисельного типу підрахунку кількості лівих Leftbrnum та правих Rightbrnum братів поточного вузла; функція обчислення глибини поточного вузла Depth. Дерево в цілому характеризують функція Degrad логічного

типу для перевірки на виродженість, яка набуває значення true, якщо дерево вироджене, і false - у протилежному випадку, та функція обчислення висоти дерева Height . Реалізована також функція Find(c) /c - вхідний параметр символьного типу/, яка повертає вказівник на перший по обходу вузол з навантаженням c.

Для спискового зображення додатково реалізовані функції Son(s,p) (Leftson(s,p), Rightson (s,p)), які перевіряють, чи є s сином /лівим або правим сином/ вузла p,і повертають значення логічного типу. Параметри s та p повинні бути типу вказівника ptr.

Операції зміни структури дерева - це вставлення піддерева Insrt(i,t) та його видалення Delt(i). За операцією Insrt(i,t) / i - вхідний параметр цілочисельного типу, t - вхідний параметр типу treeArr для векторних і ptr - для списокових зображень/ вставляється піддерево, що задається параметром t , корінь якого стає i-м сином поточного вузла. За операцією Delt(i) /i - вхідний параметр цілого типу/ відбувається видалення i-го сина поточного вузла. Обидві ці операції майже ідентичні для лужкового і бездужкового зображень, а для врахування відмінностей введені процедури корекції вставлення-видалення, які є внутрішніми і не можуть бути використані з поза меж модуля. Роль цих процедур полягає у вставлянні чи стиранні символів, що обмежують піддерево.

Використання засобів ООП дало змогу спростити деякі алгоритми. Завдяки тому що всі операції ведуться тільки з поточним вузлом, а процедура Gotonode робить неможливим некоректне переміщення по дереву, вдалося уникнути більшості перевірок на коректність аргументів операцій, які можуть бути досить трудомісткими.

Усі операції були реалізовані через чотири базові. Цей набір операцій різниться від того, який пропонується як мінімальний оцінний базис у праці [1]. При розгляді дерев як абстрактних структур даних у базовий набір входять: перехід до заданого вузла, пошук піддерева, рух уверх; перехід до наступного вузла. Проте використання як базової операції руху до наступного вузла виявилося практично незручним, тому за базові були обрані перехід до заданого вузла, пошук піддерева, рух вверх та рух униз по дереву.

Важливим результатом є те, що для лужкового і бездужкового зображень більшість операцій виявилися спільними. Різняться лише операції пошуку, руху вверх та деякі частини операцій вставлення-видалення. Решта операцій за рахунок властивостей наслідування й поліморфізму є спільними для обох зображень.

I. К о с т и в О.В. Разработка и исследование оценочных базисов прикладного сложностного анализа вычислений над деревьями: Автореф. дис.,...., канд. физ. мат. наук. Львов, 1985. 197 с.
2. Н а г а о М., К а т а я м а Т., У э м у р а С. Структуры и базы данных. М., Мир, 1986. 198 с.

Стаття надійшла до редколегії 01.03.95

УДК 519.6

Б.О.Попов, М.Ф.Лемех

УЗАГАЛЬНЕНА ОБМІННА ТЕОРЕМА ДЛЯ НАЙКРАЩОГО ЧЕБИШОВОГО НАБЛИЖЕННЯ

Розглянемо задачу найкращого чебишового наближення функції $f(x)$ з вагою $w(x)$ на проміжку $[\alpha, \beta]$ функцією від многочлена:

$$\Phi_m(x,A)=\varphi\Big(\sum_{j=0}^{m} a_j x^j\Big),\ x\in[\alpha,\beta],\ A=\{a_j\}_{j=0}^{m},$$

У працях [1,2] доведені теорема існування та характеристична теорема. Далі вважатимемо, що функції $f(x)$, $w(x)$ і $\Phi_m(x,A)$ задовольняють умови цих теорем.

Для розв"язання даної задачі застосовуємо метод чебишових ітерацій [1,2] з використанням одного з алгоритмів заміни точок альтернансу [1].

При застосуванні методу чебишових ітерацій до задачі найкращого чебишового наближення функцією від многочлена, виникає необхідність розв"язувати нелінійну систему рівнянь.

У статті запропонована наближена схема розв"язання задачі найкращого чебишового наближення функцією від многочлена - побудована задача найкращого чебишового наближення многочленом, розв"язок якої є асимптотичним до розв"язку задачі найкращого чебишового наближення функцією від многочлена. Показаний зв"язок цієї задачі найкращого чебишового наближення многочленом з обмінними теоремами для найкращого чебишового наближення. Наведені результати розв"язання тестових задач та їх аналіз.

Запишемо систему рівнянь, яка виникає при черговій ітерації для розв"язання задачі найкращого чебишового наближення функцією від многочлена: