УДК 655.287+ 519.6

І. В.Андріїв

ОСОБЛИВОСТІ ПОБУДОВИ ТА ПОДАННЯ АЛГОРИТМІВ КОМП'ЮТЕРНОГО ОПРАЦЮВАННЯ ТЕКСТУ

Питанням побудови алгоритмів присвячено багато праць у вітчизняній, і, особливо, зарубіжній науково-технічній літературі. Доволі широкого розповсюдження набули ці роботи в останні десятиріччя, коли бурхливий розвиток обчислювальної техніки спонукав до необхідності, з одного боку, шукати нові шляхи для більш ефективного розв'язання відомих задач, а з

другого. – спричинився до появи задач, постановка та реалізація яких у „докомп'ютерну епоху" не мала ні практичного, ні наукового підгрунтя. До таких задач можна віднести автоматизацію додрукарських видавничо-поліграфічних процесів для випуску книжкових, газетних, журнальних видань та іншої друкованої продукції. Основою новітніх поліграфічних технологій у цьому напрямку стали комп'ютерні видавничі системи – сучасні комплекси інформаційного та програмного забезпечення, побудовані на базі алгоритмічних моделей опрацювання текстової, графічної та образотворчої інформації.

На перший погляд видається, шо створення алгоритмів і програм опрацювання тексту – процес нескладний і доступний будь-якому спеціалістові, який володіє правилами побудови алгоритмів і мовою програмування, знайомий з технологічними вимогами до складальних процесів. Уся складність, однак, полягає в тому, що в технологічних вимогах звернено увагу на те, що дозволено або заборонено робити, але майже нічого не сказано, як це здійснити та якими засобами і методами досягти результату, особливо в ситуаціях, що взаємно виключають одна одну. Тобто маємо простий перелік вимог, виконання і дотримання яких приводить до потрібного результату. Однак у реальній ситуації не все так просто. Так, наприклад, реалізація заборони закінчувати п'ять рядків підряд переносами слів може призвести до неприпустимого збільшення величини проміжків між словами. До такого ж негативного результату можна прийти при виконанні вимоги щодо закінчення рядків прийменником, який містить одну букву. Отже, у технологічних вимогах відсутнє головне – спосіб забезпечення правил формування тексту, або, іншими словами, алгоритм процесу їх дотримання та реалізації.

Існують різні методи побудови алгоритмів. Вони стосуються в основному задач, які оперують з числовими даними на вході і результатами такого ж типу на виході. Це так звані інженерні задачі, які розв'язуються методами обчислювальної математики або чисельними методами. До них належать: чисельне розв'язування алгебраїчних і трансцендентних рівнянь; розв'язування звичайних диференціальних рівнянь та диференціальних рівнянь у частинних похідних; розв`язування задач на власні значення і задач оптимізації.

Алгоритми опрацювання текстової інформації належать до алгоритмів управління. В основу їх покладено метод моделювання станів і процесів переходу інформації з одного стану в інший. Розв'язання деякої частини текстових задач виражається математичними моделями, тому у відповідних алгоритмах потрібно дотримуватися логіки дій, закладених у цих моделях. Додаткові умови, обмеження, типи даних та структура і частково зміст результату також слідують із математично формалізованої постановки задач. В результаті одержується моделюючий алгоритм з порівняно незначною кількістю логічних умов та відповідних віток переходів.

У більшості ж задач опрацювання тексту переважає складна й розгалужена логіка, побудована на сукупному аналізі різних ознак символів та їх кодів, команд і параметрів поліграфічного оформлення тексту, умов і вимог технологічних інструкцій, які не вписуються в строгі математичні залежності. Одночасно треба пам'ятати, що формування тексту, особливо верстання складних сторінок (з огляду на варіанти розміщення в них різних елементів), є процесом творчим та індивідуальним, незважаючи на загальні правила, яких слід при цьому дотримуватися. Аналіз цих сумісних факторів дозволяє виробити певні підходи та методи, які можуть бути основою для проектування алгоритмів опрацювання тексту. В результаті функціонування таких алгоритмів створюються спеціальні масиви інструкцій і паспортів сформованого по горизонталі і вертикалі тексту, що ідентифікують, відповідно, форматовані рядки та організовані сторінки; виробляються різні ознаки, генеруються команди управління вивідною секцією (наприклад, фотоскладальною чи лазерною). Внаслідок створюється цифровий опис майбутньої книги, газети, журналу – комп'ютерна модель реального об'єкта, що є відображенням складної логіки перетворень, які здійснюються над текстовою інформацією.

Після уточнення вимог до системи, визначення її можливостей і переліку функцій, розроблення загальної структури, вибору методів реалізації функцій і мови програмування створюються достатні передумови для побудови алгоритмів. Одночасно визначаються вимоги, виконання яких можна здійснити в автоматичному режимі. При цьому треба враховувати доцільність реалізації певної функції в тому чи іншому алгоритмі, час, затрачений на її виконання, структуру інформації та її кодове зобра-

ження, кількість повних прочитань тексту програмами, логічну послідовність його формування. Весь комплекс поліграфічного опрацювання здійснюється над текстом, записаним у внутрішніх кодах системи, тобто засобами її внутрішнього алфавіту.

Технологічні вимоги не завжди однозначні, часто суперечливі між собою. Це вимагає певних зусиль для вироблення спеціальних варіантів і способів їх реалізації. В результаті аналізу генеруються додаткові умови і правила, шо забезпечують оптимальний вихід з подібних ситуацій. Так, наприклад, для ліквідації переносів, розміщених у п'яти рядках підряд, можна знехтувати забороною закінчувати рядок прийменником, який складається з однієї букви, або переформувати абзац з іншими параметрами пробілів між словами. Подібне вирішення пов'язане з усуненням першого рядка абзацу в кінці сторінки. Для цього пропонується два способи: втягнути чи витягнути рядки в абзацах, числові характеристики яких не суперечать відповідним задачам, або змінити вертикальні розміри проміжних елементів на сторінці при їх наявності. До переліку подібних ситуацій можна додати вимоги, що стосуються правил прикривання текстом рядків рубрик, ілюстрацій і т. д.

Одночасне використання декількох способів формування тексту може призвести до порушення однієї з вимог через виконання іншої. Один з можливих варіантів виходу з цієї ситуації передбачає наступні дії: задається перелік вимог, виконання яких взаємопов'язані і деяким чином впливають одна на одну; кожній із таких вимог присвоюється певний числовий коефіцієнт пріоритетності, згідно з яким вимога одержує відповідну „вагу"; визначається пряма або обернена залежність між числом (коефіцієнтом) і вагою вимоги. Результат виконання вимог ставиться в залежність від вагових коефіцієнтів. Запропоновані варіанти повинні бути в достатній мірі аргументовані та обгрунтовані.

Алгоритми, як послідовність дій розв'язання задачі, правил (команд) виконання деякої обчислювальної процедури або сукупність елементарних тактів обробки дискретних повідомлень (даних), повинні бути певним чином описані та зафіксовані. Для цього використовуються різні засоби та способи, а саме: формалізована математична модель; дегалізований словесний опис; схема передач управління (блок-схема); керуючі таблиці; алго-

ритмічні мови високого рівня; псевдомови. Кожний із них застосовується в залежності від конкретних потреб і реальної ситуації.

Найдоступнішим, на перший погляд, видається спосіб запису алгоритму звичайною мовою. Однак ця зовнішня простота зникає при заглибленні в деталі варіантів реалізації, повторень і логічних розгалужень. Для одержання закінченого в певній мірі алгоритму потрібні відповідний рівень знань, глибоке розуміння проблеми і проникнення в специфіку її розв'язання, володіння правилами і способами побудови складної логіки функціонування алгоритму та вміння використовувати їх для відображення реального процесу перетворення даних засобами словесної моделі. Інколи алгоритми, зображені в такому вигляді, передують наступним формам їх подання і, як правило, не можуть бути кінцевим продуктом розробки, а тільки результатом окремого етапу алгоритмування задач.

Подібна ситуація характерна для деяких задач комп'ютерного опрацювання тексту, початкові варіанти розв'язання яких неможливо подати іншим способом. У цьому випадку словесно формулюється перелік вимог, які складають суть процесу комп'ютерного підготування видання, його загальну логіку, структуру та зміст результату. Найчастіше мінімальний рівень словесного опису компілюється з математичними моделями алгоритму, що можливо у випадку формалізованого зображення елементарних структур даних і процедур їх опрацювання.

Досконаліший спосіб – це запис алгоритму спеціальною мовою, так званою алгоритмічною. В такому вигляді алгоритм придатний для публікацій або зберігання в спеціальних фондах алгоритмів та програм і подається у вигляді тексту, записаного на алгоритмічній мові.

Мова, що використовується для запису алгоритмів, повинна бути формальною. Це означає наявність кінцевої множини символів (одночасно знаків і їх змісту), котрі об'єднуються між собою за допомогою спеціальних правил, які називаються синтаксисом мови. Деякій, заздалегідь вибраній, сукупності символів або окремому слову, надається певний зміст або значення. Це семантика мови. Особливістю формальних мов є взаємна однозначність їх синтаксису і семантики, однозначність речень. Вхідні дані для алгоритму описуються тією ж формальною мовою.

У комп'ютерних видавничих системах (КВС) відповідність між структурою (синтаксисом) вхідної інформації та її змістом (семантикою) досягається вибором або розробленням пристрою підготування і введення даних з таким асортиментом знаків (формальною вхідною мовою), який ініціюється відповідним способом зображення та кодування клавіатури. Така вхідна мова повинна забезпечувати задання всіх необхідних конструкцій тексту та керуючої інформації (команд), які б задовольняли вимоги алгоритмів. Ці вимоги стосуються мінімізації набору символів для кодування тексту і команд, простоти і надійності задання команд та їх параметрів, наявності засобів оперативного виправлення помилок при кодуванні, відновлення неправильно заданих команд.

Використання пристроїв (клавіатури) з іншим набором знаків не обов'язково веде до заміни всіх алгоритмів. Модульна ієрархічна структура програмного забезпечення КВС, а також використання внутрішнього (системного) алфавіту опису даних, незалежного від кодів пристроїв введення і виведення, дозволяють обійтися заміною тільки тих програмних модулів, які пов'язані з переведенням вхідного подання даних у системне і наступним його перекодуванням у коди вивідного пристрою.

Для деяких задач опрацювання тексту досить ефективним є використання псевдомов. Вони можуть бути активним замінником блок-схем. На відміну від мови програмування для псевдомови не потрібний компілятор, щоб перетворити вхідний опис алгоритму в робочу програму. Псевдомова, як правило, не тільки містить в собі елементи мови програмування для керування набором команд, але й суміщає їх з описовими коментарями тих обчислень, які повинні бути виконані.

Деякі проблеми, пов'язані з вибором мови програмування для реалізації алгоритмів опрацювання тексту, стосуються характеру задач і структури текстових даних. У наступному викладі наведемо математичні залежності, що дозволяють здійснити вибір мови програмування з кількісних характеристик програм.

У роботі [3] наведені рівняння, що характеризують об'єм V, рівень L та потенційний об'єм V^* програми. Зауважимо тільки, що об'єм програми залежить від числа всіх операторів та операндів, які появляються в даній реалізації, тобто від загального словника програми. Тоді мінімальний об'єм програми вважається

потенційним. На підставі того, що $L = V^* \cdot V^{-1}$ (для ідеальної реалізації $L = 1$), робимо висновок, що у випадку переведення алгоритму з однієї мови реалізації на іншу рівень програм із збільшенням її об'єму зменшується в тій же пропорції. З другого боку, якщо мова залишається незмінною, то до аналогічного зменшення рівня програми L призводить ріст її потенційного об'єму. Тобто добуток LV залишається незмінним для будь-якої мови. Його називають рівнем мови

$$R = LV^*. \tag{1}$$

Перетворимо останній вираз:

$$R = LV^* = L(L \times V) = L^2V. \tag{2}$$

Взявши довільний показник b, одержимо

$$R = L^b \times V. \tag{3}$$

Прологарифмувавши обидві частини, матимемо

$$\ln R = b \ln L + \ln V \tag{4}$$

або після певних замін

$$X = a + bY, \tag{5}$$

де $X = \ln V$; $a = \ln R$; $Y = -\ln L$.

На основі порівняльної оцінки величин „об'єм – рівень програми" та „мова – вхідний вибір" Холстед [3] одержав експериментальне значення $b = 1.987$, близьке до заданого формулою (1). Відзначено, що чим більший розмір блока (ділянки програми між сусідніми операторами переходу), тим вищий рівень мови програмування.

Вибір мови програмування (за Холстедом) для алгоритмів опрацювання тексту має допоміжне значення. Суть задач, структури текстових даних і логіка здійснюваних над ними перетворень диктують свої умови та вимоги до засобів і методів розв'язання проблеми комп'ютерного підготування видань і розробки відповідних алгоритмів.

Проміжним варіантом опису алгоритму (між математичною моделлю і поданням його алгоритмічною мовою) може бути графічна модель алгоритму у вигляді блок-схеми, або так звана структурна схема програми. Вона зображається у вигляді визначеної сукупності плоских геометричних фігур і спеціальних графічних символів, з'єднаних між собою лініями.

Розрізняють блок-схеми різних рівнів деталізації. На початковій стадії проектування КВС формується узагальнена блок-схема, яку назвемо системною або логічною схемою. Вона є основою творення функціональної схеми, яка визначає основні функції структурних частин системи та зв'язки між ними. В таку схему повинні бути внесені інформаційні масиви, а разом з тим показана загальна логіка здійснюваних над ними перетворень. І, нарешті, для кодування алгоритму (програмування) розробляється детальна блок-схема, в якій кожний блок втілює в собі один або декілька операторів вибраної мови програмування.

Рівень деталізації блок-схеми, її повнота залежать не тільки від ступеня завершеності алгоритму, але й від рівня кваліфікації проектувальників. Слід, однак, пам'ятати, що блок-схема повинна бути побудована таким чином, щоб у ній чітко виражалися логіка роботи, можливість оцінки повноти реалізованих функцій та передачі для кодування (програмування) будь-якій групі спеціалістів. Розв'язанню цих задач сприяє, окрім того, достатня кількість коментарів у схемі.

Для опису алгоритму виділяють спільні, найчастіше вживані елементи, які називають базовими конструкціями проектування модулів. Серед них розрізняють послідовність, умову (вибір варіанту) та повторення (цикл). Незважаючи на зовнішню простоту базових конструкцій, блок-схеми в алгоритмах опрацювання тексту, побудовані з їх використанням, можуть бути доволі складними. Звичайно, у добре структурованих системах, доведених до певного оптимального рівня модульності, складність схем має деяку межу. Вона залежить від якості, глибини і, як наслідок, рівня проектування програмного забезпечення систем.

Методи опрацювання даних або способи реалізації алгоритмів можуть бути різними і залежать від змісту задачі, організації вхідних і вихідних даних, вимог до тривалості опрацювання та об'ємів пам'яті, режимів виконання задач. Узагальнюючи, виділимо такі основні класи керуючих конструкцій, які використовуються при опрацюванні даних: послідовність (що базується на повторенні); паралельні обчислення (з використанням співпрограм); довільне опрацювання (із застосуванням паралельних обчислень).

Повторення в послідовному методі обробки використовує дві форми: ітерацію і рекурсію. Ітерація визначає повторення

процесу, в якому результати одного або декількох кроків обчислень є вхідною інформацією для наступного кроку. Найчастіше ця циклічна процедура закінчується при досягненні певної межі, або після того, як її результати перестають змінюватися. Ітераційний процес є основою багатьох чисельних методів.

Розрізняють m-кроковий ітераційний процес, в якому нове значення одержується на основі m попередніх, і m-кроковий послідовний процес – нове значення залежить від останніх значень [2], тобто

$$x^{(k+1)} = G_k(x^{(k)},\ldots, x^{(k-m+1)}). \qquad (6)$$

Ітераційний процес є стаціонарним, якщо функція G_k у виразі (6) не залежить від k, тобто нове значення обчислюється з використанням попереднього значення за тією ж формулою. Так, наприклад, стаціонарний ітераційний процес описується співвідношенням

$$x^{(k+1)} = \frac{\left(x^{(k)} + \frac{1}{x^{(k)}}\right)}{2}. \qquad (7)$$

Рекурсія, яка належить до конструкції повторення, означає процес визначення або вираження функції, процедури чи розв'язку задачі через них же. В теорії програмування найчастіше розглядаються рекурсивні підпрограми, тобто такі, які викликають самі себе. Таке звертання здійснюється, як правило, через умовний оператор, бо інакше процес буде нескінченним.

Іноді одна і та ж процедура розрахунку може бути реалізована ітераційним або рекурсивним способами. Перевагу надають саме тому з них, який дозволяє економити пам`ять або (і) час обчислень.

Для одночасного доступу до всіх елементів невпорядкованого масиву та виконання операцій на різних вітках програми використовується паралельне опрацювання.

Аналогічний, але тільки стуктурний, паралелізм обчислень реалізується за допомогою співпрограм. Звичайні паралельні процеси починають виконуватися одночасно, а програмні – відповідно до ходу виконання головної програми і підпорядкованої їй підпрограми, що сукупно складають співпрограму.

Викладений матеріал дозволяє більш обгрунтовано розв'язати задачу побудови й опису алгоритмів комп'ютерного

опрацювання текстової інформації, що визначають рівень і суть програмного забезпечення комп'ютерних видавничих систем, а в результаті якість друкованої продукції.

1. Сеньківський В.М., Андріїв І.В. Систематизація та використання засобів автоматизованого опрацювання текстової інформації // Кваліологія книги: Зб.наук. праць. Львів. 2000. 2. Толковый словарь по вычислительным системам/ Под ред. В.Иллингоуорта: Пер. с англ. М., 1990. 3. Холстед М.Х. Начала науки о программах: Пер. с англ. М., 1981.

Стаття надійшла до редколегії 28.01.2000