

Дисциплина
«РАЗРАБОТКА ЭФФЕКТИВНЫХ АЛГОРИТМОВ»

Лекция 3

КОНСТРУКТИВНЫЕ АЛГЕБРЫ И МОДЕЛИ ВЫЧИСЛЕНИЙ

Лектор: Михаил Васильевич Ульянов,
muljanov@mail.ru, 8 916 589 94 04

Универсальная и конструктивная алгебры

«Универсальная алгебра» — раздел математики, изучающий алгебраические системы. В рамках этого изложения под алгеброй (универсальной) понимается *совокупность* непустого множества O (носитель) с заданным на нем набором операций S : $A = \langle O, S \rangle$.

Рассмотрим поле действительных чисел $A = \langle \mathbb{R}^1, \{+, *\} \rangle$. Можем ли мы утверждать, что существуют алгоритмы для операций $+$ и $*$ над объектами носителя? Увы — нет! Операция $\sqrt{2} + \sqrt{3}$ не алгоритмизируема в силу бесконечности числа десятичных знаков для иррациональных чисел и невозможности их записи в любой мыслимой памяти компьютера.

Под *конструктивной алгеброй* будем понимать алгебру, в которой операции алгоритмизируемы и существует механизм реализации операций R , выполняющий алгоритмы операций над объектами носителя — $A = \langle O, S, R \rangle$.

Формализация модели вычислений

Для какого подмножества действительных чисел R^1 мы можем гарантировать существование алгоритмов (в смысле Э. Л. Поста, т.е. финитного 1-процесса, доставляющего решение общей проблемы) для арифметических операций? Это ограниченное подмножество рациональных чисел Q . При этом мы ограничиваем как числитель, так и знаменатель дроби (p, q) : $|p| < N_1$, $|q| < N$.

Тогда модель вычислений (грубо, мы еще её уточним) есть совокупность:

$$M = \langle Q, S, R \rangle$$

ограниченного подмножества рациональных чисел, множества операций над этими объектами и механизма реализации, снабженного алгоритмами для выполнения операций из S .

К сожалению, полученная конструктивная алгебра является частичной, т.е. не для всех элементов из Q результат возвращается в носитель.

Модель вычислений как частичная алгебра

Рассмотрим в качестве примера следующий формат чисел в десятичной системе счисления (без знака): $0,xx*10^{yy} = xx\textcolor{red}{yy}$.

Тогда запись $55\textcolor{red}{03} = 0,55*10^3=550,00$

1. При умножении $23\textcolor{red}{76} * 43\textcolor{red}{51}$ сумма порядков равная 127 не может быть размещена в двух позициях — результат вне носителя.

2. При умножении $12\textcolor{red}{01}*12\textcolor{red}{01}$ мы получаем ограничение мантиссы $14\textcolor{red}{01}$, т.е. 1,4 вместо 1,44 — потеря точности при каждом умножении!

3. При сложении: $12\textcolor{red}{04} = 0,12*10^4 = 1200$ и $12\textcolor{red}{01} = 0,12*10^1=1,2$ вместо результата 1201,2 мы получим снова $12\textcolor{red}{04}!!!$ Т.е. $a+b=a$ при b не равном нулю! $12\textcolor{red}{04}+12\textcolor{red}{01}=12\textcolor{red}{04}!!!$

4 При сложении с меньшим сдвигом по порядку получим потерю точности.

Пример: $12\textcolor{red}{02}+12\textcolor{red}{01} = 13\textcolor{red}{02}$, а правильный результат 13,2, а не 13,0!

Свойства алгоритмов и модель вычислений

Рассмотрим более подробно свойства алгоритмов, определяющие компоненты модели вычислений:

— *свойство наличия данных*, обрабатываемых алгоритмом, может быть формально описано как существование некоторого множества объектов (чисел), над которыми выполняются операции, задаваемые алгоритмом;

— *свойство наличия памяти* для размещения объектов приводит к формальному введению некоторой алгебры, сигнатура которой описывает операции доступа к этим объектам. Такая алгебра называется *информационной алгеброй* модели вычислений;

— *свойство априорности операций* приводит к постулированию некоторого конечного набора операций, в дальнейшем называемых **элементарными**, которые однозначно определены и известны до формулировки алгоритма.

— *свойство существования механизма реализации* предполагает, что существует некоторый абстрактный механизм, который выполняет элементарные операции над объектами носителя и операции доступа информационной алгебры. Кроме того, предполагается, что механизм реализации распознает запись алгоритма, задающего последовательность операций, и тем самым предполагается наличие еще одной группы операций, которые определяют *последовательность* выполнения элементарных операций.

Таким образом, модель вычислений представляет собой совокупность информационной алгебры с собственными операциями доступа к объектам носителя, механизма реализации и априорного набора элементарных операций, в терминах которых и формулируется алгоритм решения задачи.

$$M = \langle Q, I_A, S, R \rangle$$

Описание составляющих модели вычислений

Q — ограниченное подмножество рациональных чисел;

I_A — информационная алгебра (это модель ОП с операциями чтения и записи) состоящая из нумерованных или именованных ячеек памяти, предназначенных для хранения объектов носителя с доступом к этим объектам;

S — множество базовых операций модели вычислений, выполняемых механизмом реализации

$$S = S_I \cup S_D \cup S_P \cup \{\text{stop}\}, \text{ где}$$

S_I — операции механизма реализации с информационным носителем, т.е. операции информационной алгебры I_A (доступ к объектам носителя);

S_D — операции сравнения и обработки информационных элементов;

S_P — операции управления последовательностью выполнения элементарных операций модели вычислений — условные и безусловные переходы;

Команда «stop» приводит к останову механизма реализации.

R — механизм реализации (абстрактный процессор с алгоритмами реализации операций конструктивной алгебры) в виде абстрактной модели, представляющей собой, например, композицию операционного и управляющего автоматов;

Мы остаемся в рамках классического подхода к способу записи алгоритма в элементарных операциях модели вычислений — элементарные операции выполняются последовательно одна за другой, пока не будет выполнена операция управления — перехода на некоторую другую операцию в записи исходного алгоритма. Это базовые принципы фон-Неймановской архитектуры (1949 год) в части организации вычислительного процесса — процедурная парадигма программирования.

2.3 Машина с произвольным доступом к памяти

Модель вычислений для машины с произвольным доступом к памяти (RAM) является естественной попыткой приблизить формализм алгоритма к среде его реализации. В этой модели механизм реализации приближен к реальному компьютеру за счет включения возможности хранения чисел в ячейках памяти и введения операции доступа по адресу. Формализм машины с произвольным доступом был введен с целью моделирования реальных вычислительных машин и получения оценок сложности вычислений. Отметим, что Д. Кнут рассматривает и использует еще более приближенную к реальному компьютеру модель вычислений — машину MIX.

Структура машины с произвольным доступом к памяти. Модель вычислений машины с произвольным доступом к памяти включает в себя следующие структурные компоненты:

— *данные*, обрабатываемые машиной, являются словами фиксированной длины в алфавите $\{0, 1\}$, интерпретируемые как числа в некотором фиксированном формате;

— *память* под размещение данных является памятью с произвольным доступом. Она состоит из ячеек, каждая из которых может хранить слово данных и обладает адресом, по которому может быть осуществлено обращение к этой ячейке. Ячейка с номером 0 называется сумматором;

— *элементарные операции* — множество операций машины с произвольным доступом к памяти состоит из операций с памятью, включая операцию адресации, арифметических операций, сравнений и переходов на другие операции.

— *форма записи* — операции записываются в порядке их выполнения построчно, команды переходов указывают номер или имя строки. Таким

образом, программа (запись алгоритма) представляет собой последовательность пронумерованных или именованных операций;

— *механизм реализации модели* — абстрактный процессор, выполняющий априорный набор базовых операций на основании записи алгоритма, выполненной в принятой форме записи.

Операции в машине с произвольным доступом к памяти. Элементарными операциями в машине с произвольным доступом к памяти, выполняемыми за формальную единицу времени, являются:

- запись в ячейку памяти по адресу;
- чтение из ячейки памяти по адресу;
- арифметические операции с числами;
- сравнения чисел с последующим переходом на другие команды;
- безусловный переход на другую команду;

Модель вычислений для языка процедурного программирования высокого уровня

В целях удобства анализа компьютерных алгоритмов, запись которых приближена к языку процедурного программирования высокого уровня, необходимо ввести соответствующую модель вычислений. Переход к этой модели связан с тем, что классический процедурный подход остается в настоящее время наиболее широко используемым при программной реализации алгоритмов.

Эта модель вводится как некоторое *расширение* машины с произвольным доступом к памяти в части набора операций. В дальнейшем будем считать, что объектами хранения в информационном носителе являются битовые слова фиксированной длины, обращение к которым производится по их адресу в символической записи.

Механизм реализации, аналогичный классическому последовательному процессору фон-Неймановской архитектуры, предполагает расположение программы и данных в информационном носителе. При этом считается, что элементарными операциями такого процессора являются операции, коррелированные с основными операторами процедурного языка высокого уровня.

Элементарные операции механизма реализации:

- простое присваивание: $a \leftarrow b$;
- одномерная индексация: $A[i] : (\text{адрес } (A) + i * \text{длина слова})$;
- арифметические операции: $\{ *, /, -, + \}$;
- операции сравнения: $a \{ <, >, =, \leq, \geq \} b$;
- логические операции: $(l1) \{ or, and, not \} (l2)$;

Замечания по набору операций:

— опираясь на идеи структурного программирования, из набора базовых операций исключается команда перехода, поскольку ее можно считать связанной с операцией сравнения в конструкции ветвления или цикла по условию;

— операции доступа к простым именованным ячейкам памяти для получения операндов включаются в результативные операции обработки этих операндов;

— операции индексации элементов массивов данных вынесены в отдельные элементарные операции с целью возможного согласования временных оценок программных реализаций;

— конструкции циклов не рассматриваются, т. к. могут быть сведены к указанному набору базовых операций.

Литература к лекции 3

- [1.1] Глушков В. М., Цейтлин Г. Е., Ющенко Е. Л. Алгебра. Языки. Программирование. — Киев: Наукова думка, 1978. — 318 с.
- [1.2] Белоусов А. И., Ткачев С. Б. Дискретная математика: Учеб. для вузов / Под ред. В. С. Зарубина, А. П. Крищенко. — М.: Изд-во МГТУ им. Н.Э. Баумана, 2001. — 744 с.
- [1.3] Грэхем Р., Кнут Д., Паташник О. Конкретная математика. Основание информатики: Пер. с англ. — М.: Мир, 1998. — 703 с.
- [1.4] Гасанов Э. Э., Кудрявцев В. Б. Теория хранения и поиска информации. — М.: Физматлит, 2002. — 288 с.
- [1.5] Алексеев В. Е., Таланов В. А. Графы и алгоритмы. Структуры данных. Модели вычислений: Учебник. — М.: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2006. — 320 с.
- [1.6] Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов: Пер. с англ.: — М.: Мир, 1979. — 546 с.
- [1.7] Кнут Д. Э. Искусство программирования, том 1. Основные алгоритмы, 3-е изд.: Пер. с англ. — М.: Издательский дом «Вильямс», 2002. — 720 с.