Тема 1. «Основы компьютерной алгебры»

Лабораторная работа

Компьютерная обработка информации: модели, методы, средства[[1]](#footnote-2) [[2]](#footnote-3)

Обработка информации — получение одних «информационных объектов» (структур данных) из других путём выполнения некоторых алгоритмов.

Исполнитель алгоритма — абстрактная или реальная (техническая, биологическая или биотехническая) система, способная выполнить действия, предписываемые алгоритмом.  
В современной информатике основным исполнителем алгоритмов является ЭВМ.

Выделяют числовую и нечисловую обработку.   
При числовой обработке используются такие объекты, как переменные, векторы, матрицы, многомерные массивы, константы и т.д.   
При нечисловой обработке объектами могут быть файлы, записи, поля, иерархии, сети, отношения и т.д.

В зависимости от формы представления обрабатываемой информации вычислительные машины делятся на три больших класса:

* Цифровые вычислительные машины (ЦВМ), обрабатывающие информацию, представленную в цифровой форме;
* Аналоговые вычислительные машины (АВМ), обрабатывающие информацию, представленную в виде непрерывно меняющихся значений какой-либо физической величины (электрического напряжения, тока и т.д.);
* Гибридные вычислительные машины (ГВМ), содержащие как аналоговые, так и цифровые вычислительные устройства.

С точки зрения реализации на основе современных достижений вычислительной техники выделяют следующие виды обработки информации:

* Последовательная обработка, применяемая в традиционной фоннеймановской архитектуре ЭВМ, располагающей одним процессором;
* Параллельная обработка, применяемая при наличии нескольких процессоров в ЭВМ;
* Конвейерная обработка, связанная с использованием в архитектуре ЭВМ одних и тех же ресурсов для решения разных задач, причем если эти задачи тождественны, то это последовательный конвейер, если задачи одинаковые – векторный конвейер.

Основные процедуры обработки данных:  


* Создание данных, как процесс обработки, предусматривает их образование в результате выполнения некоторого алгоритма и дальнейшее использование для преобразований на более высоком уровне.
* Модификация данных связана с отображением изменений в реальной предметной области, осуществляемых путем включения новых данных и удаления ненужных.
* Контроль, безопасность и целостность направлены на адекватное отображение реального состояния предметной области в информационной модели и обеспечивают защиту информации от несанкционированного доступа (безопасность) и от сбоев и повреждений технических и программных средств.
* Поиск информации, хранимой в памяти компьютера, осуществляется как самостоятельное действие при выполнении ответов на различные запросы и как вспомогательная операция при обработке информации.
* Поддержка принятия решения является наиболее важным действием, выполняемым при обработке информации. Широкая альтернатива принимаемых решений приводит к необходимости использования разнообразных математических моделей.
* Создание документов, сводок, отчетов заключается в преобразовании информации в формы, пригодные для чтения как человеком, так и компьютером. С этим действием связаны и такие операции, как обработка, считывание, сканирование и сортировка документов.

Основные этапы решения задач с помощью компьютера:

1. Постановка задачи: сбор информации о задаче; формулировка условия задачи; определение конечных целей решения задачи; определение формы выдачи результатов; описание данных.
2. Анализ и исследование задачи, модели: анализ существующих аналогов; анализ технических и программных средств; разработка математической модели; разработка структур данных.
3. Разработка алгоритма: выбор метода проектирования алгоритма; выбор формы записи алгоритма (блок-схема, псевдокод и др.); выбор тестов и метода тестирования; проектирование алгоритма.
4. Программирование: выбор языка программирования; уточнение способов организации данных; запись алгоритма на выбранном языке программирования.
5. Тестирование и отладка: синтаксическая отладка; отладка семантики и логической структуры; тестовые расчёты и анализ результатов тестирования; совершенствование программы.
6. Анализ результатов решения задачи и уточнение в случае необходимости математической модели с повторным выполнением этапов 2—5.
7. Сопровождение программы: доработка программы для решения конкретных задач; составление документации к решенной задаче, математической модели, алгоритму, программе, по их использованию.

Для решения задач используют:

* Метод логического вывода, основанный на технике доказательств, называемой резолюцией и использующей опровержение отрицания (доказательство "от противного");
* Метод структурной индукции, основанный на построении дерева принятия решений для определения объектов из большого числа данных на входе;
* Метод эвристических правил, основанных на использовании опыта экспертов, а не на абстрактных правилах формальной логики;
* Метод машинной аналогии, основанный на представлении информации о сравниваемых объектах в удобном виде, например, в виде структур данных, называемых фреймами.

Среди средств разработки информационных приложений .можно выделить следующие основные группы:

* Традиционные системы программирования;
* Инструменты для создания файл-серверных приложений;
* Средства разработки приложений "клиент-сервер";
* Средства автоматизации делопроизводства и документооборота;
* Средства разработки Интернет;
* Средства автоматизации проектирования приложений.

Структуры данных в компьютерной алгебре[[3]](#footnote-4)

Структурой данных называется совокупность множеств { M1, M2, ...MN } и совокупность отношений { P1, P2, ...PR }, определённых над элементами этих множеств: S = { M1, M2, ...MN; P1, P2, ...PR }.

Бинарное отношение, задающее массив – орграф. Структура данных линейна, если орграф не содержит циклов и может быть изображен в виде одной линии.

Диаграммы отношений в структурах данных:

* Отношение следования (элементы множеств – вершины, отношения следования - стрелки)
* Отношение «иметь имя» (обеспечивает доступ к элементам множеств в терминах алгоритма – a [ i ])
* Отношение «иметь значение» (обеспечивает функциональные преобразования данных – val [ i ])

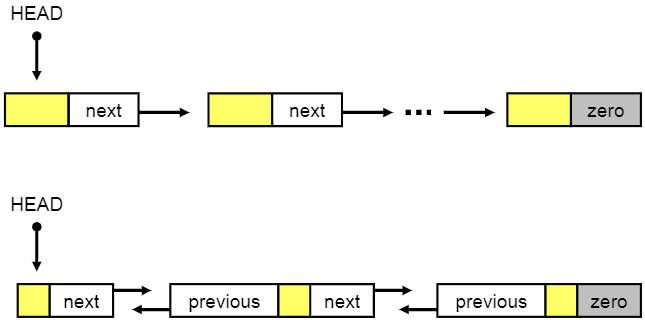
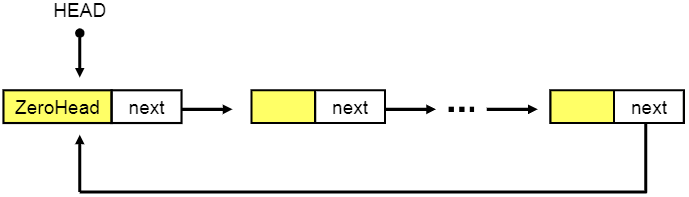
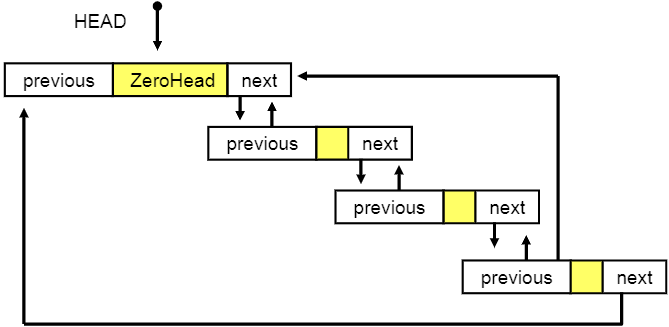
Экземпляром структуры данных называется совокупность:  
ES = { Mai, V, P, val},  
где Mai – множество элементов ai;  
V – множество значений;  
P – множество отношений следования;  
val – отношение «иметь значение».

Схемой структуры данных называется совокупность:  
SS = { Mai, P },  
гдеMai – множество элементов ai;  
P – множество отношений следования.

Одной SS может соответствовать множество ES. Алгоритм реализуется над схемой, а конкретные вычисления(преобразования) по алгоритму производятся над экземплярами.

Списки

Типы списков:

* Линейные списки: одно- и двух- связные  
  
* Циклический список: односвязный  
  
* Циклический список: двухсвязный  
  
* Рекурсивные списки  
  Рекурсивным называется список, элементами которого могут являться списками. Рекурсивные списки способны представлять данные любого уровня структурной сложности. Элементами рекурсивного списка могут быть не списки.

Основные операции

Операции над структурами данных:

* Создание и уничтожение структуры данных;
* Поиск элемента данных в структуре;
* Обновление структуры данных: вставка нового и удаление старого элемента;
* Обход структуры данных с выполнением определённых, наперёд заданных действий.

Операции над списками:

Пусть L = (A1, A2, A3, ...Ai, ...An) – список. Тогда определены следующие операции:

1. Базовые
   1. Создание нулевого списка: L = ()
   2. Получение 1-го элемента (головы) списка: A1 для списка L
   3. Получение остатка списка (переход по ссылке к следующему элементу): (A2, A3, ...Ai, ...An) для списка L
   4. Конкатенация (слияние) двух списков L1 и L2: L = (L1, L2)
2. Дополнительные
   1. Получение следующего элемента Ai+1, если известен предыдущий Ai
   2. Вставка нового элемента B после элемента Ai, т.е. получение из списка L = (A1, A2, A3, ...Ai, Ai+1, ...An) нового списка M = (A1, A2, A3, ...Ai, B, Ai+1, ...An)
   3. Удаление элемента, следующего за элементом Ai, т.е. получение из списка L = (A1, A2, A3, ...Ai, Ai+1, ...An) нового списка M = (A1, A2, A3, ...Ai, Ai+2, ...An)

Базовые типы данных

1. Числа (целые (короткие, длинные), рациональные, алгебраические, комплексные).
2. Математические выражения (арифметика, функции, производные, интегралы, матрицы, уравнения)

Типы представлений

1. МАССИВЫ (разрядность представления чисел – постоянная), (тип представления – не масштабируемое), (способ доступа к элементу – прямой (по индексу))
2. ПОСЛЕДОВАТЕЛЬНОСТИ (разрядность представления чисел – переменная), (тип представления – масштабируемое), (способ доступа к элементу – последовательный (по указателям))
3. СПИСКИ (разрядность представления чисел – переменная), (тип представления – масштабируемое), (способ доступа к элементу – последовательный (по указателям)), (способ изменения разрядности – встроенный)

Системы компьютерной алгебры: достижения и перспективы[[4]](#footnote-5)

Система компьютерной алгебры – прикладная программа для символьных вычислений, т. е. выполнения преобразований и работы с математическими выражениями в аналитической (символьной) форме.

Классы СКА по функциональному назначению:

* СКА общего назначения (Решение задач для большинства основных разделов символьной математики.   
  Примеры: Maxima, Axiom, Maple, Mathematica, Sage, Yacas)
* Специализированные СКА (Решение задач для одного или нескольких смежных разделов символьной математики.   
  Примеры: теория групп – GAP тензорная алгебра – Cadabra)

Перспективные направления развития

1. Расширение состава встроенных и программируемых типов математических объектов:
   1. Принадлежность математического объекта СКА к встроенным должна определяться не случайной практической необходимостью (СКА ранних поколений), а ролью в иерархической системе математических абстракций (СКА Axiom).
   2. Способность к созданию расширений (объектных, структурных, функциональных и т.п.) СКА должна поддерживаться интерфейсом (желательно с помощью объектноориентированного, специализированного языка программирования) (СКА Maple, СКА Mathematica).
   3. Интеграция ядра и расширений СКА должна быть прозрачной для любого пользователя СКА (цель пользователя – решение прикладной задачи, а не организация взаимодействия вычислительных модулей).
2. Интеграция СКА с другими компьютерными системами:
   1. Связь с программами числовой обработки
   2. Генерация текста программ вычислений
   3. Связь с текстовыми процессорами
3. Унификация и объектная ориентация интерфейса пользователя:
   1. Для унификации пользовательский интерфейс СКА должен иметь те же функциональные возможности, что и интерфейсы других сред программирования и проектирования (настройка параметров, редактирование объектов, отладка проектов и т.п.).
   2. Для объектной ориентации необходима реализация специальных классов объектов, представляющих алгебраические и другие абстрактные математические категории (тождества, многообразия, исчисления и т.п.).
   3. Для образовательных и рекламных целей требуется наличие инструментальных средств создания интерактивных документов (анимационная графика, панели управления и т.п.).
4. Программирование символьных вычислений произвольной сложности:
   1. Увеличение количества встроенных в СКА библиотек шаблонов пользовательских приложений для различных предметных областей (СКА Maple).
   2. Использование в качестве языка реализации СКА - функционально расширяемого языка программирования (LISP), обеспечивающее не только неограниченный рост сложности вновь создаваемых приложений, но и совершенствование базовых объектов и алгоритмов аналитических вычислений (СКА Mathematica).
5. Ускорение работы СКА:
   1. Постоянное совершенствование способов символьного представления математических объектов и алгоритмов выполнения аналитических преобразований.
   2. Применение технологии JIT-компиляции машинных кодов для реализации наиболее трудоёмких операций компьютерной алгебры (решение дифференциальных уравнений, статистическое моделирование и т.п.)

1. <https://studfiles.net/preview/1970335/page:38/> [↑](#footnote-ref-2)
2. <https://studme.org/59732/informatika/obrabotka_informatsii> [↑](#footnote-ref-3)
3. <http://docplayer.ru/57654492-Kompyuternaya-algebra-kurs-lekciy-igor-alekseevich-malyshev.html> [↑](#footnote-ref-4)
4. <http://kspt.icc.spbstu.ru/media/files/2012/course/comp-algebra/CAS_L05.pdf> [↑](#footnote-ref-5)