Моделирование множеств списками.

Работа со списками

Методические указания к лабораторной работе

<u>Цель работы</u>: научиться работать со списками и применять их в моделировании.

Теоретическая справка

1. Множества и операции на множествах.

Определение 1. *Множеством называется совокупность различимых элементов, объединённых общим признаком*.

Элементами множества могут быть числа, фигуры, предметы, понятия и т.п.

Общий признак, о котором говорится в определении, относителен, поэтому одному множеству могут принадлежать, например, и числа, и фигуры.

Множества бывают *конечными* и *бесконечными*.

1.1. Способы задания множеств

Множества задаются следующими способами:

1. перечислением элементов;
2. характеристическим предикатом;
3. порождающей процедурой.

Первый способ очевиден. Второй способ использует понятие предикат.

Предикатом называется функция одной или нескольких переменных, принимающая значение «истина» или «ложь» (0 или 1). Такая функция задаёт условие вхождения объекта в множество. При выполнении данного условия объект принадлежит множеству.

Порождающая процедура формирует элементы множества, непосредственно создавая их.

1.2. Операции на множествах.

Наиболее общее определение операции на множестве основано на понятии отображения множеств.

Определение 2. Отображением множества X на множество Y называется правило, ставящее в соответствие каждому элементу множества X элементы множества Y. Число таких элементов множества Y может быть произвольным. Обозначается отображение множеств следующим образом: $X \to Y$.

Определение 3. Отображение множества A^n в множество A, $f: A^n \to A$ называется n-арной (n-местной) операцией f, заданной на множестве A. Число n называется <u>*рангом операции*</u>. Обозначение A^n соответствует декартову (прямому) произведению n множеств:

$$A^n = \underbrace{A \times A \times ... \times A}_{n} \qquad (1)$$

Определение 4. *Прямое* или *декартово* произведение двух множеств — это множество, элементами которого являются все возможные *упорядоченные пары* элементов исходных множеств.

В выражении (1) речь идёт не о парах, а об упорядоченных наборах из n элементов; каждый элемент – это само множество A.

Другими словами, выражение (1) задаёт отображение n экземпляров множества *на себя*.

На практике наиболее часто встречаются двуместные и одноместные операции на множествах. К ним относятся следующие операции.

Объединение множеств:

$$A \cup B = \{x \colon x \in A \lor x \in B\}.$$

Пересечение множеств:

$$A \cap B = \{x \colon x \in A \land x \in B\}.$$

Разность множеств:

$$A \setminus B = \{x \colon x \in A \land x \notin B\}$$

Симметрическая разность множеств:

$$A \triangle B = (A \setminus B) \cup (B \setminus A)$$

Операция *дополнение множества* $\overline{A}$ определяется следующим образом:

$\overline{A} = U \setminus A$, где U – универсальное множество. Итак, дополнение множества - это множество всех элементов универсального множества, не принадлежащих A.

Каждое множество может быть задано посредством характеристического предиката:

$$A = \{x \colon P(x)\},\ B = \{x \colon Q(x)\} \qquad (2)$$

Легко показать, что:

$$A \cup B = \{x \colon P(x) \lor Q(x)\},\ A \cap B = \{x \colon P(x) \land Q(x)\},\ A \setminus B = \{x \colon P(x) \land \neg\, Q(x)\}. \quad (3)$$

1.3. Собственное подмножество и булеан множества

Если $B \subseteq A$, но $B \neq A$, то пишут $B \subset A$ и называют *строгим подмножеством* (или *собственным подмножеством*) множества , а символ $\subset$ - символом строгого включения.

Для всякого множества A может быть образовано множество всех подмножеств множества A. Его называют ***булеаном*** множества A и обозначают 2^A

$$2^A = \{X \colon X \subseteq A\}.$$

Для булеана используют также обозначения $\mathcal{P}(A)$ $\mathcal{B}$ и $\exp(A)$.

2. Организация списков

Списки являются одной из основных структур данных. Они широко применяются в моделировании и в функциональных языках программирования, таких как LISP.

Любая сложная структура данных может быть представлена в виде некоторого (возможно, сложного и вложенного) списка. Например, *n*-мерный массив представляется в виде списка с глубиной *n*. Любой граф - дерево также может быть представлен в виде списка.

Множество элементов любой природы может быть задано списком. Для моделирования структуры, заданной на данном множестве, используются вложенные списки.

Рассмотрим примеры, иллюстрирующие основные свойства списков и приёмы работы с ними в системе *Mathematica* [4].

Списки представляют собой совокупности однотипных или разнотипных данных, сгруппированных с помощью фигурных скобок:

- { 1 , 2 , 3 } — список из трех целых чисел;
- { a , b , c } — список из трех символьных данных;
- {1, a, x^ 2} — список из разнотипных данных;
- {{a,b},{c,d}} — сложный список, эквивалентный матрице

a b
c d

- {x^2+y^2, 2*Sin [x] } — список из двух математических выражений.

Как видно из этих примеров, элементы списков размещаются в фигурных скобках — открывающей { и закрывающей }. Списки могут быть с вложениями из списков — так получаются многоуровневые списки. Двухуровневый список задаёт матрицу.

С помощью списков представляются множественные данные — массивы.
Списки могут быть заданы различными способами. Также как и множества, списки могут быть заданы перечислением элементов, характеристическим предикатом и порождающей процедурой.

Будем различать явное задание списков и построение списков как результат определённых действий.

Пример явного задания списков:

```
{a, b, c, d}
List[a, b, c, d]
```

Оба варианта эквивалентны. Применение функции List [] задаёт <u>*полную форму выражения*</u>, которое в данном случае задаёт список.

Любые действия в системе *Mathematica* выполняются с помощью вызова соответствующих функций.

Проверить *функциональный состав выражения* можно с помощью функции FullForm[].

Задание списков двумя другими способами – с помощью предикатов и порождающей процедурой – имеет особенности в системе *Mathematica*.

1. Должна быть определена функция – генератор списка.

В следующем примере функция Range генерирует список из 4-х натуральных чисел:

```
In[1]:= Range[4]

Out[1]= {1, 2, 3, 4}
```

2. Порождающая процедура для списков использует функцию – генератор и формирует логические условия создания элементов спсика.

Пример:

```
In[1]:= f[a, Sequence[b, c], d]

Out[1]= f[a, b, c, d]
```

Программное обеспечение и оборудование для выполнения работы

Работа выполняется на современном персональном компьютере. Необходимое программное обеспечение:

- *Система Mathematica*

Порядок выполнения работы

1. Ознакомиться с теоретической справкой. Ознакомиться с литературой [1] (раздел 4.1-4.4).
2. Создать сложный список, состоящий из фамилий и имён студентов вашей группы. Каждая пара элементов данного списка – фамилия и имя – задаётся подсписком.
3. Ответить на контрольные вопросы и выполнить задания.

Контрольные вопросы и задания

1. Дайте определение бесконечному множеству.
2. Дайте определение подмножеству.
3. Почему симметрическая разность множеств названа симметрической?
4. Доказать, что из выражений (2) следуют выражения (3).
5. Используя функцию `Sort,` упорядочить список группы по первому и по второму элементам его подсписков.
6. Запрограммировать все рассмотренные операции на множестве в виде списка группы.
7. Для заданного множества построить его булеан. Воспользоваться системой помощи *Mathematica* и английскими значениями используемых терминов.

Литература и информационные ресурсы

1. Новиков Ф.А. Дискретная математика для программистов. – СПБ, Питер, 2001. -304 с.
2. Дьяконов В. П. *Mathematica* 5.1/5.2/6. Программирование и математические вычисления. – М.: ДМК-Пресс, 2008. – 576 с.
3. Богатырёв М.Ю. Прикладное моделирование в системе *Mathematica*. Основы работы с системой: Учеб. пособие по спец. 071900 «Информационные системы в технике и технологиях». – Тула, ТулГУ, 2003. – 176 с.
4. Система *Mathematica*: http://www.wolfram.com/