Лабораторная работа № 3
по дисциплине «Теория вычислительных процессов»

Алгоритмическая система Поста

1. Основные сведения из теории

Алгоритмическая система Поста оперирует цепочками символов, и базируется на введенном американским математиком Эмилем Постом понятием формальной системы.

1.1. Понятие формальной системы

Теория ФС рассматривает только синтаксические свойства изучаемых объектов - слов в некотором абстрактном языке – т.е. математические объекты в ФС понимаются как последовательности символов в некотором фиксированном алфавите, а операции над математическими объектами – операции над символами.

Формальной системой называется кортеж:

$FS = < A, A_1, R >$, где

- A – алфавит;

Будем обозначать все слова в алфавите A как A^*

- $A_1 \subseteq A^*$ - аксиомы FS или множество правильно построенных выражений в алфавите A;
- R – конечное множество вычислимых отношений $R_i(\alpha_1, ..., \alpha_n, \beta)$ на множестве A^*, называемых правилами вывода

Пример FS как Машины Тьюринга.

$$M = < Q, A, k_0, P >$$

$$FS = < A, A_1, R >$$

$$\underbrace{Q, A,}_{A} \underbrace{K_0}_{A_1}, \underbrace{P}_{R}$$

1.2. Система Туэ.

Система подстановок или полусистема Туэ – определяется как ФС через

1. алфавит A;
2. конечное множество подстановок вида $R_i(\alpha_i, \beta_i) = \{\alpha_i \to \beta_i\}$,где α_i и β_i - слова в A; причем $\alpha_i \to \beta_i$ интерпретируется, как правило вывода R_i специальным образом из слова γ выводимо слово δ $\gamma \succ \delta$, $\forall(\gamma, \delta \in A^*)$ по правилу R_i, если слово δ получается из γ путем подстановки слова β_i вместо какого-нибудь вхождения слова α_i в слово γ.

Определение Выводом слова β из α в полусистеме Туэ называется цепочка $\alpha \mapsto \varepsilon_1 \mapsto \varepsilon_2 \mapsto \ldots \mapsto \varepsilon_n \mapsto \beta$ и обозначается $\alpha \Rightarrow \beta$

Определение. Ассоциативное исчисление или система Туэ – это формальная система, определяемая некоторым алфавитом A и конечным множеством соотношений

$\alpha_i \leftrightarrow \beta_i$,

каждое из которых понимается как пара подстановок

$\alpha_i \to \beta_i$ (левая подстановка)

$\beta_i \to \alpha_i$ (правая подстановка)

Замечание. Если $\alpha \mapsto \beta$, то верно $\beta \mapsto \alpha$.

Отношение $\leftrightarrow$ в соотношении(подстановке) является отношением эквивалентности обладающее свойствами

$\forall(\alpha, \beta, \gamma \in A^*)$

- рефлексивность $\forall(\alpha)(\alpha \leftrightarrow \alpha)$
- транзитивность $\forall(\alpha, \beta, \gamma)\,([(\alpha \leftrightarrow \beta)\,\&\,(\beta \leftrightarrow \gamma) \Rightarrow (\alpha \leftrightarrow \gamma)])$
- симметричность $\forall(\alpha, \beta)[(\alpha \leftrightarrow \beta) \Rightarrow (\beta \leftrightarrow \alpha)]$

Применение ассоциативного исчисления. Поскольку выводимость здесь – отношение *эквивалентности*, то задачей анализа измерительной информации (задача контроля состояния технических объектов) может служить выявление *эквивалентных* исправных/неисправных (штатных/нештатных) технических состояний контролируемого объекта.

Пример (ассоциативного исчисления).

$$A = \{a, b, c\}$$

$$R = \{abc \leftrightarrow abd, k_1$$

$$abd \leftrightarrow add, k_2 \quad \text{команды изменения состояний}$$

$$add \leftrightarrow ddd, k_3\}$$

$abc \Rightarrow ddd$

$ddd \Rightarrow abc$

1.3. Определение канонической системы Поста

Определение. Каноническая система Поста определяется как кортеж

$FS_p = < A, X, A_1, R >$, где

1. A – собственный алфавит (алфавит констант)
   $A = \{a_1, ..., a_m\}$;
2. X – алфавит переменных
   $X = \{x_1, ..., x_n\}$;
3. A_1 – конечное множеством аксиом в алфавите $A \cup X$;
4. R – конечное множеством продукций (или правил вывода)
   $\gamma_1, ...\gamma_n \to \delta$.

Определение. Слово α получается применением некоторой аксиомы ω, в том случае, если вместо переменных в ω подставляются слова из A^*.

Определение. Слово α непосредственно выводимо из слов $\alpha_1,...,\alpha_n$ применением продукции R_i с n-посылок если найдется такая подстановка слов вместо переменных в R_i, которая посылки в R_i превратит в $\alpha_1,...,\alpha_n$, а заключение R_i - в слово α .

Определение. Последовательность слов называется доказательством в канонической системе FS_p, если каждое слово в этой последовательности:

- либо результат применения аксиомы
- либо непосредственно выводимо из предыдущих слов последовательным применением некоторой продукции системы.

Определение: слово называется доказуемым (или теорией) если оно является последним словом некоторого доказательства.

Пример (формальной системы Поста).
М.Т. – вычисляла значение вычислимой функции по некоторому значению ее аргумента. Но вычислимую функцию можно задать и прямым перечислением множества ее значений.
Например, множество всех нечетных чисел (1,3,5,…) также может быть получено не только с помощью МТ , а с помощью FS_p :

$$A = \{1\},$$
$$X = \{x\},$$
$$A_1 = \{1\};$$
$$R = \{x \to x11\}.$$

Результат выполнения: $\{1,111,11111\}$.

Для порождения степени двойки можно использовать следующее правило: $R = \{x \to xx\}$.

1.4. Вычисления с использованием формальной системы Поста

Вычисления в формальной системе Поста как и любые другие вычисления носят дискретный характер. Вычисление в ФС Поста – последовательный вывод с использованием правил (последовательное применение правил). Вычисление происходит до тех пор, пока возможно применить какое-либо правило.

1.4.1. Применение правил.
- 1.4.1.1. Правила выбираются по одному;
- 1.4.1.2. Каждое правило анализируется на возможность его применения;
- 1.4.1.3. Если ни одно правило применить нельзя, вычисления прекращаются.
- 1.4.1.4. Правила должны быть составлены таким образом, что на каждом шаге вычисления, было возможно применить лишь одно правило.
- 1.4.1.5. Когда найдено правило, которое возможно применить вычисляются значения переменных входящих в левую часть правила.
- 1.4.1.6. На основе вычисленных значений переменных формируется строка по правой части правила.
- 1.4.1.7. После того как значения переменных вычислены, в исходной строке происходит замена вычисленной левой части правила на вычисленную правую часть правила.

1.5. Пример вычисления с использованием формальной системы Поста

Рассмотрим вычисление арифметической функции $z + \frac{x}{y}$.

Пусть z=5, x=9, y=3, зададим исходную строку следующим образом:
11111+111111111/111=

$$A = \{1, +, /, =\},$$
$$X = \{x, y\},$$
$$A_1 = \{1\};$$
$$R = \left\{ \begin{matrix} (1) x/x = \to /x = 1, \\ (2) x + /y = \to = x \end{matrix} \right\}.$$

Процесс вычисления будет выглядеть следующим образом:
Выбираем первое правило и анализируем его на применимость.
Первое правило применить возможно.
Выбираем второе правило и анализируем его на применимость.
Второе правило применить невозможно, поскольку в анализируемой строке отсутствует последовательность символов +/
Применяем первое правило

Результатом применения первого правило станет строка:
11111+111111/111=1
Проверив правила на применимость, используем первое правило.
Результат
11111+111/111=11
Проверив правила на применимость, используем первое правило.
Результат
11111+/111=111
Проверив правила на применимость, убеждаемся то правило №1 применить невозможно, а правило №2 применимо
В результате применения правила №2 получаем строку
=11111111
Проверяем правила на применимость
Ни одно из правил применить нельзя.
Вычисления прекращаются
Полученная строка содержит корректный результат вычисления арифметической функции.

2. Задание на лабораторную работу

В данной лабораторной работе требуется:

- Построить формальную систему Поста FS_P, реализующую вычисление заданной арифметической функции.
- Написать программу на языке высокого уровня имитирующую (эмулирующую) вычисления на основе выводимости в формальной системе Поста.
- Программа должна работать на любых входных данных из заданного множества.
- Программа должна удовлетворять предъявляемым требованиям.

3. Требования к программе:

Программа должна корректно обрабатывать ошибочные входные данные;
Контроль однозначности заданных правил проводить не требуется;

Следующие входные данные должны считываться программой из файла

- A – собственный алфавит
- X – множество переменных
- A_1 – множество аксиом
- R – конечное множество продукций (или правил вывода)

Результат работы программы должен содержать:

- Файл с результатами вычисления
- Сообщение на экране о следующих событиях:
    - Вычисление прошло успешно;
    - В ходе вычисления произошла ошибка:
        - Найден символ не входящий в алфавит;
        - Найдена переменная не входящая в заданное множество переменных;

4. Требования к входным данным:

- Все входные данные должны содержаться в одном файле.
- В файле элементы ФС должны быть заданы в следующем виде:
  A ={<элемент алфавита 1>, <элемент алфавита 2>,…,<элемент алфавита N>};
  X ={<переменная 1>, <переменная 2>, … ,< переменная K>};
  A_1 ={<аксиома 1>, <аксиома 2>, … , <аксиома M>};
  R ={<правило 1>,
  <правило 2>,
  …,
  <правило L>}

5. Требования к выходным данным:

- В файл с результатами вычисления на каждое применение правила выводятся следующие строки:
  исходная строка;
  применяемое правило;
  результат применения правила.

6. Содержание отчета

- Цель работы;
- Постановка задачи;
- Листинг программы на языке высокого уровня с комментариями;
- Содержимое входного файла согласно заданию;
- Примеры результатов выполнения:
    - Примеры должны содержать корректные и некорректные входные данные;
    - Содержимое выходных файлов и сообщения выводимые на экран.
- Вывод.

7. Варианты заданий:

Задание на удовлетворительно

1.$x+2$

2.$x+3$

3.$x+4$

4.$x+5$

5.$x+7$

6.x_1+x_2

7.$x_1+x_2+x_3$

8.x_1+x_2+1

9.x_1+x_2+3

10.x_1+x_2+4

11.x_1+x_2+5

12.$x_1+x_2+x_3+1$

13.$x_1+x_2+x_3+2$

14.$x_1+x_2+x_3+4$

Задание на хорошо

15.$x-2$

16.$x-3$

17.$x-y$

18.x_1+x_2-y

19.$x-y+1$

20.$x-1+y$

21.x_1+x_2-1

22.$y-x-1$

23.x_1+x_2+y

24.x_1-x_2-y

Задание на отлично

30.$2x$

31.$3x$

32.$x_1 x_2$

33.x / y

34.$x / 4$

35.$2x_1 + x_2$

36.$x_1 x_2 + y$

37.$x_1 x_2 - 3$

38.$x_1 x_2 - y$

39.$3x - y$

40.$y_1 - y_2 x$

41.$y_2 - \frac{x}{y_1}$

42.$5x + y$

43.$x / 10$

44.$z + \frac{x}{y}$

8. Список литературы по теме работы:

1. Алферова З.В. Теория алгоритмов. Учебное пособие для вузов. – М. Статистика, 1973 – 164с.
2. Городецкий В.И. Прикладная алгебра и дискретная математика.
   Часть 2. Формальные системы логического типа. – МО СССР, 1986 – 200с.
   Часть 3. Формальные системы логического типа. – МО СССР, 1987 – 177с.