МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»

Кафедра теории функций и стохастического анализа

ОТЧЕТ ПО ПРОИЗВОДСТВЕННОЙ (БАЗОВОЙ) ПРАКТИКЕ

студента 4 курса 451 группы
направления 38.03.05 — Бизнес-информатика

механико-математического факультета
Чайковского Петра Ильича

Место прохождения: завод "Тантал"
Сроки прохождения: с 29.06.2019 г. по 26.07.2019 г.
Оценка:

Руководитель практики от СГУ

доцент, к. ф.-м. н. ______________ Н. Ю. Агафонова

Руководитель практики от организации

ведущий программист ______________ Д. Э. Кнутов

Саратов 2019

Тема практики:«Правила оформления курсовых и дипломных работ»

СОДЕРЖАНИЕ

1 Постановка задачи

Цель работы — изучение основных свойств бинарных отношений и операций замыкания бинарных отношений.

Порядок выполнения работы:

1. Разобрать основные определения видов бинарных отношений и разработать алгоритмы классификации бинарных отношений.

2. Изучить свойства бинарных отношений и рассмотреть основные системы замыкания на множестве бинарных отношений.

3. Разработать алгоритмы построения основных замыканий бинарных отношений.

2 Теоретические сведения по рассмотренным темам с их обоснованием

Определение. Подмножества декартова произведения $A \times B$ множеств A и B называется *бинарными отношениями* между элементами множеств A, B и обозначаются строчными греческими буквами: $\rho, \sigma, \ldots, \rho_1, \rho_2, \ldots$.

Определение. Бинарное отношение $\rho \subset A \times A$ называется:
1. *рефлексивным*, если $(a,\, a) \in \rho$ для всякого $a \in A$;
2. *симметричным*, если $(a,\, b) \in \rho \implies (b,\, a) \in \rho$;
3. *антисимметричным*, если $(a,\, b) \in \rho$ и $(b,\, a) \in \rho \implies a = b$;
4. *транзитивным*, если $(a,\, b) \in \rho$ и $(b,\, c) \in \rho \implies (a,\, c) \in \rho$.

Символом Δ_A обозначается тождественное отношение на множестве A, которое определяется по формуле:

$$\Delta_A = \{\, (a,\, a) \,|\, a \in A \,\}.$$

Тогда бинарное отношение $\rho \subset A \times A$ является:
1. *рефлексивным*, если $\Delta_A \subset \rho$;
2. *симметричным*, если $\rho^{-1} \subset \rho$;
3. *антисимметричным*, если $\rho \cap \rho^{-1} \subset \Delta_A$;
4. *транзитивным*, если $\rho\rho \subset \rho$.

Определение. Бинарное отношение ρ на множестве A называется:
1. *отношением эквивалентности (эквивалентностью)*, если оно рефлексивно, симметрично и транзитивно.
2. *отношением порядка (порядком)*, если оно рефлексивно, антисимметрично и транзитивно.
3. *отношением квазипорядка (квазипорядком)*, если оно рефлексивно и транзитивно.

Определение. Множество Z подмножеств множества A называется *системой замыканий*, если оно замкнуто относительно пересечений, т.е. выполняется:

$$\bigcap B \in Z \quad \forall B \subset Z.$$

В частности, для $\varnothing \subset Z$ выполняется $\cap\varnothing = A \in Z$.

Лемма 1. О системах замыканий бинарных отношений. На множестве $P(A^2)$ всех бинарных отношений между элементами множества A следующие множества являются системами замыканий:

1. Z_r - множество всех рефлексивных бинарных отношений между элементами множества A.

2. Z_s - множество всех симметричных бинарных отношений между элементами множества A.

3. Z_t - множество всех транзитивных бинарных отношений между элементами множества A.

4. $Z_{eq} = Eq(A)$ - множество всех отношений эквивалентности на множестве A.

Множество Z_{as} всех антисимметричных бинарных отношений между элементами множества A не является системой замыканий.

Определение. *Оператором замыкания* на множестве A называется отображение f множества всех подмножеств $P(A)$ в себя, удовлетворяющее условиям:

1) $X \subset Y \implies f(X) \subset f(Y)$;

2) $X \subset f(X)$;

3) $(f \circ f)(X) = f(X)$

для всех $X, Y \in P(A)$. Для подмножества $X \subset A$ значение $f(X)$ называется *замыканием* подмножества X.

Лемма 2. О замыканиях бинарных отношений. На множестве $P(A^2)$ всех бинарных отношений между элементами множества A следующие отображения являются операторами замыканий:

1) $f_r(\rho) = \varrho \cup \Delta_A$ - наименьшее рефлексивное бинарное отношение, содержащее отношение $\rho \subset A^2$,

2) $f_s(\rho) = \varrho \cup \varrho^{-1}$ - наименьшее симметричное бинарное отношение, содержащее отношение $\rho \subset A^2$,

3) $f_t(\rho) = \bigcup\limits_{n=1}^{\infty} \rho^n$ - наименьшее транзитивное бинарное отношение, содержащее отношение $\rho \subset A^2$,

4) $f_{eq}(\rho) = (f_t \circ f_s \circ f_r)(\rho)$ - наименьшее отношение эквивалентности, содержащее отношение $\rho \subset A^2$.

3 Результаты работы

3.1 Алгоритм определения свойства рефлексивности.

Описание алгоритма определения свойства рефлексивности.

Вход: список смежности бинарного отношения ρ.

Выход: строка «Бинарное отношение является/не является рефлексивным.» и bool значение **true** или **false**.

Метод: для каждого элемента a, находящегося в бинарном отношении ρ (с некоторыми элементами b_i), просматривается его список смежности. В этом списке смежности ищется сам элемент a. Алгоритм прекрщает свою работу, если был найден элемент a, список смежности которого не содержит a, или, если для всякого элемента a его список смежности содержит a.

Псевдокод алгоритма определения свойства рефлексивности.

```
isReflexive(binaryRelation)
{
  for element in binaryRelation
  {
    flag = false;

    for subelement in element:
      if (subelement == element):
        flag = true;

    if (!flag)
      return false;
  }

  return true;
}
```

Листинг 1: Псевдокод алгоритма.

Код программы, реализующей алгоритм определения свойства рефлексивности.

```
bool isReflexive(map<int, set<int>> binaryRelation)
{
  bool isReflexive = false;
  set<int> ::iterator it;
  int i;

  for (auto element : binaryRelation)
  {
    isReflexive = false;
    it = element.second.begin();

    for (; it != element.second.end(); ++it)
      if (element.first == *it)
      {
        isReflexive = !isReflexive;
        break;
      }

    if (!isReflexive)
      break;
  }

  cout << "\n" <<
    (isReflexive ? "\nBINARY RELATION IS REFLEXIVE.\n" :
                   "\nBINARY RELATION IS NOT REFLEXIVE.\n");
  return isReflexive;
}
```

Листинг 2: Код программы.

Результат тестирования программы определения свойства бинарного отношения.

Для демонстрации работы программы рассмотрим два произвольных бинарных отношения ρ и δ, на первом из которых свойство рефлексивности выполняется, а на втором нет.

Сгенерируем 8 пар элементов, соответствующих бинарному отношению ρ. Получаем следующие пары:

```
INPUT THE NUMBER OF PAIRS:
8
NOW INPUT THE PAIRS:
(13; 1). (16; 68). (17; 9). (34; 98). (56; 5). (60; 42). (81; 61). (91; 57).
```

Рисунок 1 – Пары элементов, находящихся в нерефлексивном бинарном отношении.

Как видно, данное отношение не является рефлексивным. Посмотрим на выход программы:

```
INPUT THE NUMBER OF PAIRS:
8
NOW INPUT THE PAIRS:
(13; 1). (16; 68). (17; 9). (34; 98). (56; 5). (60; 42). (81; 61). (91; 57).

BINARY RELATION IS NOT REFLEXIVE.
```

Рисунок 2 – Проверка на рефлексивность не пройдена.

Выход программы совпадает с тем фактом, что отношение не является рефлексивным. Теперь рассмотрим бинарное отношение δ, на котором свойство рефлексивности выполняется. Рассмотрим 8 следующих пар:

```
INPUT THE NUMBER OF PAIRS:
8
NOW INPUT THE PAIRS:
(0; 0), (0; 1), (0; 2). (1; 0), (1; 1), (1; 2). (2; 2).
```

Рисунок 3 – Пары элементов, находящихся в рефлексивном бинарном отношении.

Как видно, отношение является рефлексивным. Посмотрим на выход программы:

```
INPUT THE NUMBER OF PAIRS:
8
NOW INPUT THE PAIRS:
(13; 1). (16; 68). (17; 9). (34; 98). (56; 5). (60; 42). (81; 61). (91; 57).

BINARY RELATION IS NOT REFLEXIVE.
```

Рисунок 4 – Проверка на рефлексивность не пройдена.

Выход программы совпадает с тем фактом, что отношение является рефлексивным.

3.2 Алгоритм определения свойства симметричности.

Описание алгоритма определения свойства симметричности.

Вход: список смежности бинарного отношения ρ. **выход:**

3.3 Описание алгоритмов построения основных замыканий бинарных отношений

3.4 Псевдокоды рассмотренных алгоритмов

3.5 Коды программ, реализуюших рассмотренные алгоритмы

3.6 Результаты тестирования программ

3.7 Оценки сложности рассмотренных алгоритмов