Федеральное агентство связи

Сибирский Государственный Университет Телекоммуникаций и Информатики

Межрегиональный центр переподготовки специалистов

# Контрольная работа

# По дисциплине: Математическая логика и теория алгоритмов.

Выполнил: Козлов Е.И.

Группа: ПБТ-82\_\_\_

Вариант: 17\_\_\_\_\_\_\_\_

Проверил: Мачикина Е. П.

Новосибирск, 2020 г

**Исчисление высказываний.**

1. **Пользуясь определением формулы исчисления высказываний проверить, является ли данное выражение формулой.**

**Решение:**

Запишем определения формулы исчисления высказываний:

1. Пропозициональная переменная есть формула.

2. Если А – формула, то (А) и  также формулы.

3. Если А и В – формулы, то , , ,  - тоже формулы.

4. Других формул нет.

Будем просматривать исходное выражение и заменять все найденные в нём формулы символом Ф. Если исходное выражение – формула, то после всех замен должен остаться один символ Ф.

Рассмотрим исходное выражение и заменим в нём все найденные пропозициональные переменные на знак Ф (формула) согласно правилу 1:

.

Согласно правилу 2 заменим  на Ф:



Согласно правилу 3 заменим на Ф:

.

Согласно правилу 2 заменим (Ф) на Ф:

.

Согласно правилу 3 заменим на Ф:

**.

Согласно правилу 2 заменим (Ф) на Ф:

**.

Согласно правилу 3 заменим на Ф:

Ф.

Больше замен провести нельзя. В итоге получили единственный символ Ф, следовательно. Исходное выражение является формулой.

1. **Записать рассуждение в логической символике и проверить правильность рассуждения методом Куайна, методом редукции и методом резолюций.**

Незнание правил дорожного движения не освобождает от ответственности в случае их несоблюдения. Для того чтобы нести ответственность нужно нарушать правила. Следовательно, знать правила нужно.

**Решение:**

Обозначим высказывания:

А – «*соблюдение ПДД*»;

В – «*ответственность за нарушение ПДД*»;

С – «*знание ПДД*».

Тогда первую гипотезу «*Незнание правил дорожного движения не освобождает от ответственности в случае их несоблюдения*» в символьном виде можно записать: .

Вторая гипотеза «*Для того чтобы нести ответственность нужно нарушать правила*» в символьном виде равна: .

Вывод «*Следовательно, знать правила нужно*» можно записать: *С*.

В общем виде рассуждение можно записать как

.

Метод Куайна. Пусть С истинно. Тогда высказывание принимает значение «истина» при любых значениях В и А.

Пусть С ложно. Если  истинно, то высказывание  также принимает значение «истина». То есть при значении В истина или при ложном В и истинном А обе аксиомы истинны.

Таким образом, исходное высказывание  не является тавтологией, то есть логически неверно.

Метод редукции. Попробуем найти такую интерпретацию, при которой высказывание примет значение «ложь». Для этого требуется, чтобы обе посылки  были истинны, а *С* ложно.

Тогда посылки принимают вид . Отсюда получаем, что при  равном «истина» высказывание принимает ложное значение. То есть при значении В истина или при ложном В и истинном А высказывание ложно.

Следовательно, существует интерпретации, при которой высказывание ложно.

То есть исходное рассуждение  логически не верно (не является тавтологией).

Метод резолюций. Представим первую посылку в виде КНФ:

.

Представим вторую посылку в виде КНФ:

.

Добавим к списку посылок отрицание вывода:

.

Для полученной системы постараемся получить пустой дизъюнкт, что будет соответствовать противоречивости системы посылок и отрицания вывода:

1)  - первая часть первая посылка;

2)  - вторая посылка.

Больше посылок вывести нельзя. Пустую посылку вывести не удалось. Следовательно, исходное выражение не является тавтологией. То есть оно может принимать ложные значения. То есть исходное рассуждение  логически неверно.

**Исчисление предикатов**

1. **Пользуясь определением формулы логики предикатов проверить, что выражение является формулой. В формуле указать свободные и связанные переменные. Привести формулу к предваренной форме.**

**Решение:**

Будем просматривать исходное выражение и заменять все найденные в нём формулы символом Ф. Если исходное выражение – формула, то после всех замен должен остаться один символ Ф.

Заменим в исходном выражении все атомарные формулы символом Ф. При этом арность одинаковых предикатных символов (в нашем случае А и В) должна совпадать:

*.*

Если Ф – формула, то ** и ** также формула (*х* - переменная). Используя данное правило, дважды проведём замену:





Если Ф – формула, то (Ф)также формула:



Если Ф – формула, то конструкция **также формула:

Ф.

Остался единственный символ Ф – исходное выражение является формулой.

Свободные и связанные переменные:

Первый предикатный символ : *х* – свободная переменная, *у* – связанная;

Второй предикатный символ : обе переменные связанные.

Преобразуем формулу к ПНФ.

Для построения предварённой нормальной формы (ПНФ) выполним следующий алгоритм:

Шаг 1. Исключим всюду логические операции импликации:

Импликаций в формуле нет.

Шаг 2. Продвинем отрицания до элементарной формулы:

Отрицаний в формуле нет.

Шаг 3. Переименуем связанные переменные.

.

Шаг 4. Вынесем кванторы влево.

.

Шаг 5. Преобразуем бескванторную матрицу к виду КНФ:

Бескванторная матрица уже представлена в виде КНФ.

Общий вид ПНФ:

**.

**Теория алгоритмов**

1. **Построить машину Тьюринга для перевода из начальной конфигурации в заключительную. На ленте МТ записаны нули и единицы, пустые ячейки содержат нули, . Проверить работу машины Тьюринга для конкретных значений *x,y*. Нарисовать граф, соответствующий построенной МТ.**
2. **Построить машину Тьюринга:**

Порядок действий для МТ. Сначала происходит движение головки вправо, вычисление количества 1 в первом слове до появления на ленте нуля. Этот нуль разделяет аргументы на ленте. После подсчета количества 1 в первом слове будет дописано или сокращено количество 1 во втором слове, после чего МТ завершит свою работу.

q1 1 → 0 R q2 – проходим первую 1 вправо;

q2 1 → 0 R q3 – проходим вторую 1 вправо;

q2 0 → 0 R q7 – переходим к шагу q7 (если кол-во 1 ≤ 2);

q3 1 → 0 R q4 – проходим третью 1 вправо;

q3 0 → 0 R q7 – переходим к шагу q7 (если кол-во 1 ≤ 2);

q4 1 → 0 R q4 – заменяем все последующие 1 на 0;

q4 0 → 0 R q5 – находим разделитель;

q5 1 → 1 R q6 – оставляем первый символ второго слова.

q6 1 → 0 R q6 – заменяем все остальные символы после первого на 0 (затираем);

q6 0 → 0 H q0 – завершаем работу машины;

q7 1 → 1 R q8 – оставляем или задаём первый символ второго слова;

q7 0 → 1 R q8 – оставляем или задаём первый символ второго слова;

q8 1 → 1 R q9 - оставляем или задаём второй символ второго слова;

q8 0 → 1 R q9 - оставляем или задаём второй символ второго слова;

q9 1 → 1 R q10 - оставляем или задаём третий символ второго слова;

q9 0 → 1 R q10 - оставляем или задаём третий символ второго слова;

q10 1 → 0 R q10 – заменяем все остальные символы на 0 (затираем);

q10 0 → 0 H q0 – завершаем работу машины.

1. **Проверить работу машины Тьюринга:**
2. *x* = 1, *y* = 2.

q11011→0q2011 → 00q711 → 001q81 → 0011q9 → 00111q10 → 00111q0.

1. *x* = 3, *y* = 5.

q1111011111 → 0q211011111 → 00q31011111 → 000q4011111 → 0000q511111 → 00001q61111 → 000010q6111 → 0000100q611 → 00001000q61 → 000010000q6 → 000010000q0.

1. **Нарисовать граф:**

1, 0 R

1, 0 R

1, 0 R

0, 1 R

0, 1 R

0, 1 R

0, 0 R

1, 0 R

1, 1 R

1, 1 R

1, 1 R

0, 0 R

1, 1 R

0, 0 R

0, 0 R

1, 0 R

0, 0 R

1, 0 R

1. **Показать примитивную рекурсивность функции f(x, y).**

**Решение:**



Для начала, докажем примитивную рекурсивность некоторых функций:

Усечённая разность примитивно рекурсивна:



Заметим, что , тогда покажем примитивную рекурсивность усечённой разности:



Теперь заметим, что функция знака примитивно рекурсивна:



Достаточно просто доказывается примитивная рекурсивность функции из слагаемых(множителей), приведём пример (прочие доказательства происходят аналогичным образом):



Тогда, рассмотрим такую характеристическую(индикаторную) функцию:

, она также примитивно-рекурсивна.

Наконец, константа примитивно рекурсивна: так как константа может быть получена в результате операций увеличения на один функции

Теперь представим изначальную функцию вот в таком виде:

Как видим, получена суперпозицией примитивно-рекурсивных функций и, таким образом, примитивно рекурсивна.