

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»
(БГТУ им. В.Г. Шухова)

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Лабораторная работа №2

по дисциплине: Математическая логика и теория алгоритмов

Тема: «Логика предикатов»

Выполнил: ст. группы ПВ-211
Чувилко Илья Романович

Проверил:
Куценко Дмитрий Александрович

Белгород 2022 г.

Задача 4.3: Определить, какие вхождения переменных являются свободными, а какие связанными в следующих формулах. Указать области действия кванторов

$$\exists x Q(x, x) \& R(f(y, x))$$

Областью действия квантора по переменной x является формула $Q(x, x)$.

В области действия квантора переменная x - связанная.

Переменная y - является свободной.

Задача 10.5: Проверить, являются ли следующие формулы тождественно истинными

$$\overline{\forall x A(x)} \leftrightarrow \exists x \overline{A(x)}$$

$$\begin{aligned} \overline{\forall x A(x)} &= \overline{\bigwedge_x A(x)} = \overline{A(a_1) \& \dots \& A(a_i) \& \dots} = \overline{A(a_1)} \vee \dots \vee \overline{A(a_i)} \vee \dots = \\ &= \bigvee_x \overline{A(x)} = \exists x \overline{A(x)} \end{aligned} \quad \text{Получим: } \exists x \overline{A(x)} \leftrightarrow \exists x \overline{A(x)}, \text{ что является тождественно истинной формулой}$$

Через двойственность кванторов: выражение "не для всех x в области определения $A(x)$ - истинно", равносильно выражению "существует x , для которого $A(x)$ - ложно"

Задача 15.2: Определить, являются ли следующие формулы общезначимыми, противоречивыми или выполнимыми в указанной интерпретации

$$\mathcal{D} = \{1, 2\}; a = 1; f(1) = 2, f(2) = 2; P(1) = 0, P(2) = 1; Q(1, 1) = Q(1, 2) = Q(2, 2) = 1, Q(2, 1) = 0$$

$$\begin{aligned} \exists x (P(x) \& Q(x, a)) &= \bigvee_x (P(x) \& Q(x, a)) = (P(1) \& Q(1, a)) \vee (P(2) \& Q(2, a)) = \\ &= (P(1) \& Q(1, 1)) \vee (P(2) \& Q(2, 1)) = (0 \& 1) \vee (1 \& 0) = 0 \vee 0 = 0 \end{aligned}$$

Формула **противоречива** на заданной интерпретации

Задание 29.4: Определить, унифицируемо ли каждое из следующих множеств. Если да, то получить наиболее общий унификатор.

$$\{Q(x, y, z), Q(u, h(v), u)\} = W$$

Начнем искать наиболее общий унификатор:

$$1) \text{ множество рассогласований } \mathcal{D} = \{x, u\}$$

$$\lambda = \{x/u\} - \text{подстановка}$$

$$W^\lambda = \{Q(x, y, z), Q(x, h(v), x)\}$$

$$2) \mathcal{D} = \{y, h(v)\}; \lambda = \{y/h(v)\}$$

$$W^\lambda = \{Q(x, y, z), Q(x, y, x)\} \Rightarrow \text{множество не унифицируемо}$$

Задание 53: В бюджете возникнет дефицит, если не повысят пошлины. Если в бюджете возникнет дефицит, то расходы на социальные нужды сократятся. Следовательно, если повысят пошлины, то расходы на социальные нужды не сократятся. Проверить это с помощью метода резолюций.

A - Пошлины не повысят

B - Дефицит возникнет

C - Расходы на соц. нужды сократятся

$$(A \rightarrow B) \& (B \rightarrow C) \rightarrow (\bar{A} \rightarrow \bar{C}) \equiv 1$$

$$\overline{(A \vee B) \& (B \vee C) \vee (A \vee \bar{C})} \equiv 1$$

$$(A \vee B) \& (B \vee C) \& A \& C \equiv 0$$

Выпишем дизъюнкты:

$$\begin{array}{l} \bar{A} \vee B \\ \bar{B} \vee C \\ \bar{A} \\ C \end{array} > \bar{A} \vee C$$

Пустой дизъюнкт получить нельзя.

Ответ: Утверждение ложно

Практическая часть:

Вариант 1: Разработать программу, способную считывать несколько формул- посылок логики высказываний и выводить на экран все формулы-следствия из этих посылок.

Для решения были взяты функции из прошлой работы.

*// Возвращает множество дизъюнкций, состоящее из переменных vars,
// согласно таблице истинности table*

```
std::set<Queue> getDisjunctionsFromTable(const TruthTable &table,
                                         const Set &vars) {
    std::set<Queue> res;
    for (const auto &row: table) {
        Queue values = row.first;
        auto var = vars.begin();
        Queue currExpr;
        while (!values.empty()) {
            Token token = values.front();
            if (token == '1')
                currExpr.push('!');
            currExpr.push(*var);
            if (std::next(var) != vars.end())
                currExpr.push('|');
            var++;
            values.pop();
        }
        res.insert(currExpr);
    }
    return res;
}
```

*// Выводит на экран все комбинации дизъюнкций для заданного СКНФ,
// т.е. формирует и выводит все следствия*

```
void printConsectaries(const std::vector<std::set<Queue>>
                       &conjunctionSubsets) {
    for (const auto &subset: conjunctionSubsets) {
        for (auto it = subset.begin(); it != subset.end(); ++it) {
            auto expr = *it;
            std::cout << '(';
            while (!expr.empty()) {
                std::cout << expr.front();
                expr.pop();
            }
            std::cout << ')';
            if (std::next(it) != subset.end())
                std::cout << '&';
        }
        std::cout << '\n';
    }
}
```

*// Получает сочетания для множества в пределах [currElement, end)
// размера subsetSize и записывает их в subsets*

```
template<typename T, typename Iter>
void getCombinations(Iter currElement, Iter end, const size_t subsetSize,
                     std::set<T> &currSubset,
                     std::vector<std::set<T>> &subsets) {
    while (currElement != end) {
        currSubset.insert(*currElement);
        if (currSubset.size() == subsetSize)
```

```

        subsets.push_back(currSubset);
    else {
        getCombinations(std::next(currElement), end, subsetSize,
                        currSubset, subsets);
    }
    currSubset.erase(*currElement);
    currElement++;
}
}

```

// Возвращает массив подмножеств множества set

```

template<typename T>
std::vector<std::set<T>> getSubsets(std::set<T> &set) {
    std::vector<std::set<T>> subsets(1);
    std::set<T> currSubset;
    for (int i = 1; i <= set.size(); i++)
        getCombinations(set.begin(), set.end(), i, currSubset, subsets);
    return subsets;
}

```

```

int main() {
    SetConsoleOutputCP(CP_UTF8);
    // Ввод посылок
    std::cout << "Введите количество посылок: ";
    size_t n;
    std::cin >> n;
    getchar();
    std::cout << "Введите посылки: ";
    std::string expr;
    for (int i = 0; i < n; ++i) {
        auto currExpr = "(" + inputExpr() + ")";
        expr += currExpr;
        if (i + 1 != n)
            expr += '&';
    }
    Queue input = stringToSequence(expr);
    try {
        // Преобразование в ОПЗ
        Queue output = infixToPostfix(input);
        printSequence(output);
        auto vars = getVariables(output);
        // Получим строки таблицы, необходимые для построения СКНФ
        auto table = deleteRowsIf(getTruthTable(output, vars),
                                [](const auto &pair) {
                                    return pair.second == '1';
                                });
        // Преобразуем строки таблицы в дизъюнкции
        auto disjunctions = getDisjunctionsFromTable(table, vars);
        // Получим все подмножества множества дизъюнкций
        auto disjunctionsSubsets = getSubsets(disjunctions);
        std::cout << "Следствия из посылок:";
        printConsectaries(disjunctionsSubsets);
    } catch (const String &err) {
        printErrorMessage(err);
        exit(1);
    }
    return 0;
}

```

C:\BGTU\BGTU\MatLogika\2lab\Code\cmake-build-debug\Code.exe

Введите количество посылок:2

Введите посылки: Формула логики высказываний: $(X \& Z) \supset (Y / Z)$

Формула логики высказываний: $X \supset Z$

$XZ \& YZ \mid \supset XZ \supset \&$

Следствия из посылок:

$(\neg X \mid \neg Y \mid Z)$

$(\neg X \mid Y \mid Z)$

$(\neg X \mid \neg Y \mid Z) \& (\neg X \mid Y \mid Z)$

Process finished with exit code 0

Вывод: в ходе работы была изучена логика предикатов и закрепились навыки решения теоретических и практических задач